University of Strathclyde

Department of Electronic and Electrical Engineering

Measurement, control and protection of microgrids at low frame rates supporting security of supply

Volume 1

Andrew Roscoe

A thesis presented in fulfillment of the requirements for the degree of Doctor of Philosophy

February 2009

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has lead to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed :   Date :

Contents

Glossary of terms

AB or *AB0*	Vector produced from a three-phase voltage or current set after application of the Clarke Transformation
AC	Alternating Current
ADC	Analogue to Digital Converter
ADI	Applied Dynamics International
AVR	Automatic Voltage Regulator
CHP	Combined Heat and Power
Clarke-FLL hybrid	The frequency, amplitude and phase measurement system proposed by this thesis
CLTF	Closed Loop Transfer Function
CPU	Central Processing Unit (of a computer or microcontroller)
CT	Current Transformer
DAC	Digital to Analogue Converter
DFIG	Dual Fed Induction Generator
DC	Direct Current
DG	Distributed Generation
DNO	Distribution Network Operator
dq or *dq0*	Vector produced from a three-phase voltage or current set after application of the Park Transformation
ETS	(EU) Emissions Trading Scheme
FIR	Finite Impulse Response (filter)
FLL	Frequency Locked Loop
GPS	Global Positioning System (>30 Satellites provide position and accurate time information with 24-hour global coverage).
IGBT	Insulated Gate Bipolar Transistor
IIR	Infinite Impulse Response (filter)
LOM	Loss Of Mains
LPF	Low Pass Filter
NDZ	Non Detection Zone (normally, in the context of loss-of-mains detection)
OLTF	Open Loop Transfer Function
PF	Power Factor
PID	Proportional plus Integral plus Differential control loops.
PLL	Phase Locked Loop
POR	Phase Offset Relay
pu	per-unit
RMS	Root Mean Squared
ROC	Renewable Obligation Certificates
ROCOF	Rate Of Change Of Frequency
RTS	Real Time Station (Multi-processor simulation and control development system from ADI)
Sa/s	Samples per second (frame rate)
SNR	Signal to Noise Ratio
THD_V	Total Harmonic Distortion (of voltage)
UPS	Uninterruptible Power Supply
VT	Voltage Transformer

Index of figures

Abstract

Increasing penetrations of distributed generation at low power levels within electricity networks leads to the requirement for cheap, integrated, protection and control systems. To minimise unit cost, algorithms for the measurement of AC voltage and current waveforms should be implemented on a single microcontroller, which also carries out all other protection and control tasks, including communication and data logging. This limits the frame rate of the major algorithms, although ADCs can be over-sampled using peripheral control processors on suitable microcontrollers. Measurement algorithms also have to be tolerant of poor power quality which may arise, even transiently, within a microgrid, battlefield, or disaster-relief scenario. This thesis analyses the potential magnitude of these interfering signals, and presents suitably tolerant architectures and algorithms for measurements of AC waveforms (amplitude, phase and frequency). These algorithms are shown to be robust and accurate, with harmonic content up to the level of 53% THD, and with the major algorithms executing at only 500 samples per second. This is achieved by the careful optimisation and cascaded use of exact-time averaging techniques, which prove to be useful at all stages of the measurements: from DC bias removal to low-sample-rate Fourier analysis to sub-harmonic ripple removal. Algorithms for three-phase nodal power flow analysis are benchmarked on the Infineon TC1796 microcontroller and require less than 8% of the 2000μs frame time, leaving the remainder free for other algorithms.

Furthermore, to optimise security of supply in a microgrid scenario, loss-of-mains must be detected quickly even when there is an accidental or deliberate balance between local active power generation and demand. The measurement techniques are extended to the detection of loss-of-mains using a new Phase Offset relay, in combination with a novel reactive power control technique to avoid the non-detection-zone. These techniques are tested using simulation, captured network transient events, and a real hardware microgrid including a synchronous generator and inverter.

1 Introduction

1.1 Distributed generation and microgrid background

The common model for western power networks within the last 50 years has been that of large systems dependent upon centralised power generation. Coal, nuclear, hydro, gas and oil-fired power stations of the multi-megawatt to multi-gigawatt scale have been built at convenient locations, and the electrical networks connected around them. The core of the UK electrical transmission grid is a 400kV and 275kV backbone which allows the electrical power to be moved with reasonable efficiency from generation to load. The coal, gas, nuclear and large hydro power stations which have provided the bulk of our electrical power are sensibly sited near to bulk fuel sources (coal mines, mountain lakes) and/or cold water supplies for cooling requirements (e.g. rivers, coastline). Some flexibility in siting relative to the fuel source locations is available for coal, gas, and nuclear power since the energy density of these fuels can justify transportation over large distances to reach the power station.

Over the coming years, it is predicted that these large centralised generators will become less dominant, and an increasing share of the generation mix will be provided by a large number of smaller scale generators, many of which will use renewable energy sources (DTI, 2007 and Ault, 2006). The reasons for this are:-

- The environmental taxes on fossil fuels and incentives for renewable energy sources. Examples of mechanisms currently active in the UK are the EU Emissions Trading Scheme (ETS) (DEFRA, 2008), and the UK Renewables Obligation which issues Renewables Obligation Certificates (ROCs).

- The poor efficiency of large power stations, due to large unrecovered heat losses.

- The gradual exhaustion of fossil fuel supplies, and the volatility of fossil-fuel prices in a global market. The UK, for example, is heavily reliant on imported natural gas from Norway and Russia. The natural gas wholesale cost increases of late 2005 provide a good example of market volatility and how vulnerable the UK is to sudden energy shortages due to the relative non-diversity of our current energy portfolio, coupled with limited UK gas storage capability. Much of our coal for power stations comes from Poland or even further afield. Even nuclear power, as proposed by the recent energy review (DTI, 2006 & 2007), does not present a long-term solution since the expected global sources of Uranium will be exhausted by approximately 2091 based upon the 2004 consumption rate (IAEA, 2007 & DTI, 2006). U_3O^8 prices more than doubled from \$10/lb in 2000 to \$21/lb in 2005, when the DTI energy review estimated the fuel cost to be 11% of the energy cost. Since

2005, the cost of U_3O^8 has risen to $72/lb at 05/01/2007 (IAEA, 2007) and then to $135/lb in July 2007. Clearly, nuclear fission represents only an expensive and short-term energy supply unless fuel reprocessing can be made much more efficient and/or the use of fast breeder reactors becomes politically acceptable.

To enable the reduction of dependence upon the diminishing fossil-fuel and nuclear-based resources, there are several solutions which can be implemented in parallel.

1. Increasing the proportion of renewable energy

2. Increasing the efficiency of thermal generation by using CHP (combined heat and power) schemes.

3. Reducing demand through taxation, real-time pricing schemes (Roscoe, 2004 & 2009c), behavioural attitudes, and the improved efficiency of equipment (EST, 2008).

General demand reduction measures are outside the scope of this thesis. Regarding the generation sources, one of the ways of achieving (1) is the installation of large windfarms at scales of up to and beyond 100MW (to date, up to 322MW on land (Scottish Power, 2006) and 520MW at sea (BBC, 2002)). Another way to achieve (1) and (2) is to install many thousands of small generators operating at the kW and MW scale, distributed within the electricity network, to replace relatively few GW-scale power stations that are retired. Such generation is commonly referred to as Distributed Generation (DG). The work of this thesis is primarily focussed at assisting the deployment of these smaller-scale generators.

The application of CHP schemes causes this distribution effect simply because generators must be sited near to the heat loads to minimise heat distribution costs. The extreme example of this is the application of domestic combined boiler/generator solutions. Another reason for the distribution of the new generators is due to the low energy density of fuels derived from renewable sources. For example, transporting of most biomass feedstock is not energy efficient over more than a few km (Cloonan, 2004), so the expectation will be for many small-scale power plants located by necessity where the resource exists. Solar, wave and wind-power installations must, by definition, be installed where the resource exists.

So, there are many drivers leading towards a network containing many thousands of small generators, some of which will be despatchable and some of which will be based upon stochastically varying resources. Most of the renewable sources will be largely

uncontrollable apart from generation curtailment, but hydroelectric and biomass/biogas plants or renewable schemes involving hydrogen storage can offer additional degrees of controllability depending upon the amount of locally stored feedstock.

The effects of this shift towards ubiquitous generation are many. One of the effects is that the risk of power outages may rise, due to winter demand peaks that cannot be met by installed renewable generation due to unfavourable weather conditions. Most of the renewable energy sources are not controllable, and it is (currently) not financially viable to store the electrical energy from wind, wave or solar installations due to the conversion losses and capital cost of adequate storage systems (Foote, 2005).

Any rise (or perceived rise) in the risk of power outages will lead to even more installations of distributed generation, by customers for whom outages, even for a few seconds, might be very expensive. During a power outage, the distributed generation can be used as an emergency generator to supply local loads in a power island.

Power networks that can be studied and controlled/protected in a localised manner can be termed microgrids. A microgrid may contain generation sources of many types, loads, electrical storage, and connection points to other microgrids or parent networks. At any time a microgrid might be islanded, connected to another microgrid, or connected to a much larger power system such as the UK national grid. Good examples of microgrid applications are:-

- Distributed generators and associated local loads within the UK distribution network
- Marine and Aeronautical power systems
- Disaster-relief power networks
- Battlefield power systems

Any electrical power system requires protection, and any generation source additionally requires control. Optimum management of a microgrid power system requires both protection and control. To research the overall goal of microgrid management, a laboratory demonstrator has been created at the University of Strathclyde. A schematic of this is shown in Figure 1-1. A significant microgrid control application has been created, which is outside the scope of this thesis (Roscoe, 2005, 2007 & 2008). While creating this application, difficulty was faced due to specific constraints and lack of established knowledge in two key areas. Without solutions to these two challenges, the effectiveness

of the microgrid control application to ensure security of supply to the local customers is significantly impaired. These two major areas are:-

1) Measurement of amplitude/phase/frequency with low frame rates within power systems experiencing poor power quality.

2) Reliable, timely detection of Loss-Of-Mains (LOM) especially when local active power generation is accidentally or deliberately balanced to local active power demand, while avoiding spurious (nuisance) tripping.
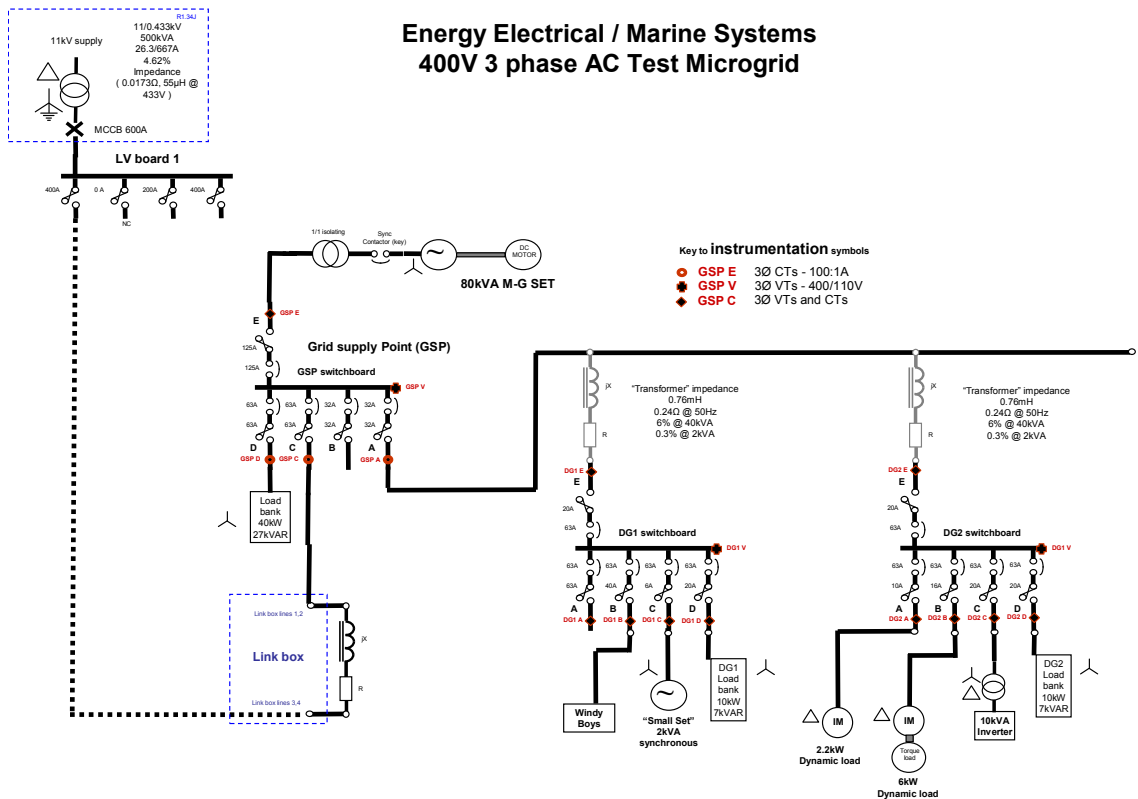
Figure 1-1 : 400V 3-phase microgrid laboratory; single-line diagram

1.2   Measurements at low frame rates

Within the context of large thermal power stations, protection and control can be implemented using large, expensive pieces of equipment, since the cost and size of the equipment is small compared to the rating and size of the power station. Within the scope of microgrid management, however, this is not true. Measurement, protection and control functions must be integrated within small, cheap pieces of hardware to become cost effective. The ideal solution is therefore for a single microcontroller to be able to manage all these tasks. The control tasks themselves (excluding those for inverter design) can easily accommodate low frame rates such as 100 Sa/s (samples per second) with a 10ms reaction time. For the measurement algorithms, however, the use of low frame rates is a

major concern since the AC waveform frequency is typically 50 Hz and this must be captured adequately by the algorithms. This thesis addresses the requirement of measuring AC signals using low frame rates down to 500 Sa/s.

The constraints on the measurements are:-

- Many measurement, protection and control algorithms need to be executed on a single microcontroller platform. The microcontroller code includes not only the required algorithms but also overheads including ADC sampling, communications and data logging. The combined execution time of the entire microcontroller code thus limits the achievable sample (frame) rate. The target frame rate used in this thesis is 500 Sa/s, equating to a frame time of 2000µs. This frame time is the lowest practical frame time currently achievable in the laboratory for an actual microgrid management algorithm that incorporates multiple measurements, together with protection and control algorithms, communication interfaces and data logging.

- Within a microgrid, power quality may be very poor (chapter 2 provides a full analysis). Measurement algorithms need to be robust enough to maintain good accuracy despite such scenarios of poor power quality.

The context of the established knowledge is that:-

- Squeezing power-system measurement algorithms onto microcontroller platforms with acceptable sample rates is not a new problem.

- Published applications have, however, focussed mainly upon squeezing relatively simple, compartmentalised, optimised algorithms onto microcontrollers, with the highest sample rate possible. Even multi-function digital relays are designed primarily to measure voltages and currents at just a single node.

- The speed and available memory within cheap microcontrollers is ever increasing. This offers new opportunities compared to methods previously proposed.

- Published academic research to date does not provide measurement algorithms which are tolerant enough to poor power quality, whilst also providing the low-latency response times required. The performance and tolerance of published algorithms is often difficult to assess due to poor graphical presentations (inappropriate scaling) within the referenced material.

The new approach adopted by this thesis is to create algorithms which can operate at very slow (and fixed) sample rates (down to 500 Sa/s, 10 samples per cycle at 50 Hz), with low

execution times, low latencies, and high accuracy.

A suitable target for execution speed is <200µs for a three-phase, nodal voltage/current/balance and power flow analysis. Making such an algorithm execute at only 500 Sa/s appears at first glance to be an unnecessary step, since such an algorithm could be clocked at 1/200µs = 5000 Sa/s which would make accuracy targets easier to achieve. However, the key point is that once such an algorithm is created, many such algorithms can be combined on a single processor, and the target frame rate still achieved. Thus, the creation of such new measurement algorithms (Roscoe, 2009), and a thorough assessment of their performance during dynamic and steady-state conditions, is a significant enabler for any microgrid control application. Additional useful by-products of this work are that the resulting algorithms can be used within multi-rate simulations to minimise simulation time, and that in future (due to microcontroller speed increases) the algorithms can be executed at higher sample rates and the accuracy will only improve from the analyses presented in this thesis.

1.3   Reliable detection of loss-of-mains

Within a microgrid scenario, to maximise security of supply, loss-of-mains (LOM) must be detected quickly and reliably. A loss-of-mains event is defined as a severing of connection to a parent network, when the local power network contains generation (and optionally load). Undetected, this condition can present risks of electrocution or damage. The LOM event must therefore be detected in a timely manner and suitable action taken. In the UK, ER G59/1 (ENA, 1991) currently forbids a distributed generator to back-feed any part of the distribution network during a LOM condition, and the LOM event must conventionally lead to a fast disconnection of the generator. However, within a customer's own private installation the detection of LOM may be used to trigger a managed transition into an islanded state, so long as no power is fed back into the public network. The ability to switch quickly to islanded mode allows a significant increase in the security-of-supply at the local level, by reducing the frequency and voltage excursions from nominal and hence the risk of local outages, even though such outages may be very short. To accomplish this effectively requires a fast and reliable method for LOM detection. Many published works present active methods for detecting this LOM condition, which rely on fast switching outputs from inverter-connected generation. For general application using rotating generators, a more passive method must be used. While methods such as monitoring dV/dt & power-factor have been proposed (Salman, 2000), methods using ROCOF (Rate of Change of Frequency) relays are the most well established and widely used. However, they suffer from two main drawbacks:-

- Inability to quickly detect the LOM condition when there is an exact or close

match between the local active and reactive power generation and demand. This is because when active and reactive power is balanced in this way, an unexpected LOM event causes only a very small (within the bounds of normal operation) change in frequency, which is not sufficient to trip LOM relays. Such a balanced of active power can be highly desirable within a microgrid, and may deliberately be targeted. This is because, when operated in this state, any subsequent deliberate transition to the islanded state results in only small frequency deviations, hence providing the highest chance that the local power island will survive the transition to islanded mode.

- Spurious tripping due to the relatively noisy measurement of ROCOF, which is a time-derivative of the measured frequency.

This thesis addresses both these issues, via two separate approaches in parallel. The first approach is the combined use of the novel accurate low sample-rate measurements with a new method to detect the LOM condition based upon estimations of the phase of the local power system relative to the parent network. This is substantially less noisy and more discriminatory than a ROCOF measurement. It can also be adapted to automatically de-sensitise itself during faults and thus avoid spurious trips. The second approach is a combination of a new design of control loop for DG power control in grid-connected mode, combined with a novel reactive power control strategy which allows successful detection of the LOM condition even when active power is exactly balanced (Roscoe, 2008b & 2009b). This works by causing a small shift in the active power balance subsequent to a LOM event and thus triggering an unstable control mode in the standard generator droop controllers. This strategy is applicable to all generators which allow control of reactive power (or power factor) without the need for power electronic devices or high frequency current injections.

1.4   Structure of this thesis

Chapter 2 describes the requirements for the measurement algorithms. These requirements encompass the required accuracy, response time (latency), tolerance to poor power quality, and constraints due to microcontroller hardware. Significant analysis is carried out to determine worst-case levels of "influence qualities" due to worst-case microgrid power quality. This accounts for such parameters as rate-of-change-of-frequency (ROCOF), harmonic content, inter-harmonics, unbalance, flicker, and instrumentation noise. The worst cases account for applicable power system standards in place today in the UK and the USA, in combination with practical assessments of whether these levels might be breached within non-standard power-system scenarios. The worst case power quality is found to be much poorer than normally experienced in the UK, although the

expectation would be that such poor power quality would only be experienced for short time periods.

Chapter 3 presents the selection and design of architectures and designs for algorithms to measure amplitude and phase, assuming that the signal frequency is known. Initially, some small building blocks are developed and characterised in isolation. The single most useful building block, which is used repeatedly throughout this thesis, is an exact-time averaging block. This block is developed from a MATLAB Simulink block, but significantly enhanced in this thesis both for mathematical accuracy and for execution speed. The use of the improved algorithm allows the creation of a useful new DC blocking algorithm with zero group delay, which offers better performance than a high-pass filter. More importantly, several architectures for the overall measurement of amplitude and phase are designed and compared. Some of these are based upon existing literature, but the best architecture for use in distribution networks is found to be a new architecture developed during chapter 3. This involves the use of 1½ cycle measurements (1-cycle exact-time integration/averaging cascaded with an extra ½-cycle exact-time averaging) and offers excellent performance even at the lowest sample rates. A selection process is presented which recommends the best architecture to be employed based upon the target scenario.

Chapter 4 begins with the design and verification of a mathematical tool which can be used to predict the likely measurement error ripples due to aliased harmonics at a range of sample rates. This shows that the potential measurement ripple at the sample rates of interest, due to certain problematic harmonics, can be large relative to the desired specification. To mitigate this problem, two solutions are applied. Firstly, a very effective, novel, adaptive ripple-removal filter is designed and tested. Secondly, a front-end 6x oversampling (3000 Sa/s) FIR notch filter is designed using standard zero-pole placement techniques; this can be used to further reduce the measurement ripple when the input signal has high levels of distortion, requiring only very small amounts of processing at the 6x oversampled frame rate.

Chapter 5 builds directly upon the outputs of chapters 3 and 4, to create a large novel algorithm (called a "Clarke-FLL hybrid") for the measurement of frequency (and amplitude and phase) within a 3-phase AC power system. Again, the cascaded use of the enhanced exact-time averaging techniques is found to be an extremely applicable technique. The Clarke-FLL hybrid is compared to seven other candidate algorithms for frequency measurement, and found to surpass them in terms of measurement accuracy and latency.

Chapter 6 presents the algorithmic design for a robust Phase Offset Relay (POR), which was initially proposed as an improved method for loss-of-mains detection in Dysko (2006), compared to traditional ROCOF and vector-shift relays. This thesis presents the first implementation of this relay in a robustly-coded form suitable for deployment on a microcontroller. The presented relay algorithm includes a new triggering subsystem which allows both trigger and trip thresholds to be set appropriately based upon known network behaviour. Also, the relay includes new software to detect balanced & unbalanced faults. When these are detected, the relay can de-sensitise itself via a temporary widening of trip setting to allow for post-fault power-system oscillations. This allows the relay to avoid spurious trips during distant and close-in network faults, providing much improved discrimination over existing LOM relays, without resorting to a complete disabling of the trip signal during such faults. This relay re-uses the novel signal processing techniques and measurement outputs developed during chapters 3-5. Substantial analysis, using both simulated and captured power system events, shows that this relay exhibits good sensitivity and discrimination.

Furthermore, analysis of power system stability is carried out in chapter 6, and combined with a new strategy for management of reactive power flow within a microgrid (Roscoe, 2008b & 2009b). This can be used to avoid the small non-detection-zone of the loss-of-mains detection algorithm, even when there is an accurate balance between locally produced active power and the local active power demand. The combination of the new control algorithm and the new loss-of-mains detection algorithm are rigorously tested using microcontroller hardware and the microgrid of Figure 1-1 (including both a synchronous generator and three-phase inverter).

Throughout this thesis, intermediate findings are highlighted where relevant. The main conclusions are summarised in chapter 7. Appendix A to Appendix H contain relevant supplementary information and additional coding details.

1.5 References for chapter 1

Ault, G. et. al. (2006). Electricity Network Scenarios for 2020. SuperGen Future Network Technologies Consortium. EPSRC/SGFNT/TR/2006-001. [Online]. [Accessed 05/08/2008]. <http://www.ensg.gov.uk/assets/dwg_pg1-p01_scenarios.pdf>

BBC (2002). Ireland takes wind power plunge. [Online]. [Accessed 05/08/2008]. <http://news.bbc.co.uk/1/hi/sci/tech/1755413.stm>

Cloonan, Currie, Roscoe & Geraghty. (2003). Biomass - Using Anaerobic Digestion. MSc group project. [Accessed 05/08/2008]. <http://www.esru.strath.ac.uk/EandE/Web_sites/03-04/biomass//index.html>

DEFRA. (2008). European Union Emissions Trading Scheme. [Online]. [Accessed 05/08/2008]. <http://www.defra.gov.uk/environment/climatechange/trading/eu/>

DTI. (2006). The energy challenge, energy review report 2006. [Online]. [Accessed 18/03/2008]. <http://www.berr.gov.uk/files/file31890.pdf>

DTI. (2007). Meeting the energy challenge, a white paper on energy. [Online]. [Accessed 05/08/2008]. <http://www.berr.gov.uk/energy/whitepaper/page39534.html>

Dysko, A., Burt, G. & Bugdal, R. (2006). Novel Protection Methods for Active Distribution Networks with High Penetrations of Distributed Generation – Year 2 Report. DTI/CDGSEE/TR/2006-7. [Online]. [Accessed 14/03/2008]. <http://www.sedg.ac.uk/DGSEE Protection Project - Year 2 Report.pdf>

ENA. (1991). ER G59/1. Recommendations for the connection of embedded generating plant to the public electricity suppliers' distribution system.

EST. (2008). Energy Saving Trust . [Online]. [Accessed 05/08/2008]. < http://www.energysavingtrust.org.uk/>

Foote, C. et. al. (2005). Ubiquitous Energy Storage. Conference on future power systems. Amsterdam, Nov 16-18 2005. Paper O10-09.

IAEA. (2007). UxC Nuclear Fuel Price Indicators. [Online]. [Accessed 05/01/2007 & 13/03/2008]. <http://www.uxc.com/review/uxc_Prices.aspx>

OFGEM. (2008). Renewables Obligation. [Online]. [Accessed 05/08/2008]. <http://www.ofgem.gov.uk/Sustainability/Environmnt/RenewablObl/Pages/RenewablObl.aspx>

Roscoe, A. J. (2004). Demand Response and Embedded Storage to Facilitate Diverse and Renewable Power Generation Portfolios in the UK. MSc thesis. [Online]. [Accessed 05/08/2008]. <http://www.esru.strath.ac.uk/Documents/MSc_2004/roscoe.pdf >

Roscoe, A. (2005). DGI test results, 28-29 June 2005. University of Strathclyde UTC document RR/URBAN/TR/2005-2.

Roscoe, A. (2007). DGI (inverter connected) test results, 30 July – 1 August 2007. University of Strathclyde UTC document RR/URBAN/TR/2007-01.

Roscoe, A. (2008). Local power control (LPC) list of functions and user manual. University of Strathclyde UTC document RR/ENERGY/TS/2008-01.

Roscoe, A. (2008b). Power flow management to avoid the non-detection of loss of mains. UK patent application 0810512.4 filed 10 June 2008.

Roscoe, A. J. & Burt, G.M. (2009). Frequency and fundamental signal measurement algorithms for distributed control and protection applications. IET Journal on generation, transmission and distribution.

Roscoe, A. J. & Burt, G.M. (2009b) (anticipated). Avoidance of the Non-Detection Zone of Passive Loss-of-Mains relays. IEEE transactions on power delivery.

Roscoe, A. J. & Ault, G. (2009c) (anticipated). Supporting high penetrations of renewable generation via implementation of real-time pricing and demand elasticity. IET Journal on renewable power generation.

Salman, S.K., King, D.J. & Weller, G. (2000). New loss of mains detection algorithm for embedded generation using rate of change of voltage and changes in power factors. Developments in Power System Protection, 2001, Seventh International Conference on (IEE) 9-12 April 2001 Page(s):82 - 85

Scottish Power. (2006). ScottishPower Awards £330m Contracts for Europe's Largest Windfarm (Whitelee). [Online]. [Accessed 05/08/2008].
<http://www.scottishpower.com/PressReleases_1070.htm>

2   System study of amplitude, phase and frequency measurement requirements within 3-phase AC microgrids

The measurement of amplitude, phase and frequency within microgrid power systems presents new problems. Traditional methods for measuring these parameters have been developed within the context of large, relatively stable power systems at high voltages where the waveforms are closely regulated and contain low levels of harmonic contamination. The new requirements for measurement algorithms within smaller power systems are analysed in the following sub-sections. The purpose of this chapter is to generate a set of requirements which such measurement algorithms will need to meet to be useful, accurate and robust within a microgrid scenario. The requirements encompass both the properties and qualities of the signals to be measured, and also the potential constraints on the measurement hardware. This may need to be substantially cheaper than existing equipment, and also to be integrated with many other software algorithms on an integrated micro-controller platform. The requirements also define suitable signals to be used as test inputs for any candidate algorithms.

2.1   Increased rates of change of frequency within microgrids

A major technical barrier to operating a small AC microgrid is the issue of inertia and system frequency stability. Within any AC power system, the frequency stability is a function of the inertia of the generators and loads, coupled with the magnitude of any load changes or generator prime mover power output changes. Restricting discussions to rotating generators for the time being, the maximum rate of change of frequency (ROCOF), in Hz/s, can be estimated for a hypothetical power system.

The per-unit inertia of a prime-mover & generator unit is given by H, in seconds, which is equal to the energy stored in the spinning unit at nominal speed divided by the nominal power rating (in VA) of the machine. Thus, if H=0.5 for a 1MVA generator, then the machine has 500kJ of stored energy when spinning at nominal speed.

H can be related to the inertia J in SI units (kgm^2) using the standard formula for stored energy in a flywheel:-

$$E = \frac{1}{2}J\omega^2$$

(2.1)

and then evaluating this at the nominal rotational speed to obtain

$$E_0 = \frac{1}{2}J\omega_0^2$$

(2.2)

Where E_0 = stored energy at nominal speed (Joules)

ω_0 = nominal rotational speed in radians/s

J = moment of inertia of prime mover plus generator, in kgm^2

Then:-

$$H = \frac{\frac{1}{2}J\omega_0^2}{S_0}$$

(2.3)

or:-

$$J = \frac{2H \cdot S_0}{\omega_0^2}$$

(2.4)

Where H = the per-unit inertia of the generator

S_0 = the nominal rating (in VA) of the generator

Per-unit inertias of generators vary, depending upon the design and size of the generator and prime mover. For example, H might be as low as <1s for small reciprocating engines coupled to synchronous generators, or as high as 10s for a large thermal GW-scale unit (Mullane, 2005).

Using these different values of H, and different sizes of power system, it is possible to perform an approximate analysis of frequency stability. The scenario to be analysed begins with a network which at some instant in time is in an equilibrium state with the sum total of all generator prime mover outputs matching the sum total of load powers. At this time the network frequency is assumed nominal, i.e. $f=f_0=2\pi\omega_0$. Then a new load is added or generator removed, which creates a generation/load imbalance. Within the immediate time following the load addition, the prime mover outputs do not change significantly, due to the response time of the governors, droop controls and throttle delays. By differentiating the standard equation for stored energy (2.1) we have:-

$$\frac{dE}{dt} = \frac{d}{d\omega}\left(\frac{1}{2}J\omega^2\right)\frac{d\omega}{dt}$$

(2.5)

$$\Rightarrow \frac{dE}{dt} = J\omega\frac{d\omega}{dt}$$

which approximates to

$$\frac{dE}{dt} = J\omega_0\frac{d\omega}{dt}$$ when the generator is close to nominal speed.

(2.6)

Note that (2.6) can be expanded into the familiar swing equation by substituting H for J and turning the energy flow from the machine dE/dt into a per-unit quantity, i.e.

$$\frac{dE}{dt} = \left(\frac{2H \cdot S_0}{\omega_0^2}\right)\omega_0\frac{d\omega}{dt}$$

(2.7)

$$\frac{\frac{dE}{dt}}{S_0} = \frac{2H}{\omega_0}\frac{d\omega}{dt}$$

(2.8)

and finally to the familiar swing equation:-

$$\frac{2H}{\omega_0}\frac{d^2\delta}{dt^2} = P_m - P_e$$

(2.9)

where P_m and P_e, and the imbalance between them (which equals $(dE/dt)/S_0$), are measured in per-unit quantities, and δ is the generator rotor angle, where of course $d\delta/dt$ is the rotational speed ω.

Returning to (2.6), this equation may be rearranged to give the rate of change of frequency by:-

$$\frac{d\omega}{dt} = \frac{1}{J\omega_0}\frac{dE}{dt}$$

(2.10)

and thence

$$\frac{df}{dt} = \frac{1}{4\pi^2 J f_0} \frac{dE}{dt}$$

(2.11)

where $f = 2\pi\omega$ and $f_0 = 2\pi\omega_0$, which reveals the rate of change of frequency (ROCOF) as a result of a power imbalance in an electrical power system with inertia.

This equation may also be expressed in terms of H by substituting (2.4):-

$$\frac{df}{dt} = \frac{f_0}{2 \cdot S_0 \cdot H} \frac{dE}{dt}$$

and thence into per-unit quantities

$$\frac{df_{p.u.}}{dt} = \frac{\Delta P_{p.u.}}{2 \cdot H}$$

(2.12)

where $\Delta P_{p.u.}$ is the load-generation power imbalance in per-unit.

To compare expected ROCOF magnitudes in the current UK national grid, and a potential microgrid, estimations can be made of representative values of J and dE/dt. Within the national grid, the generators are generally large synchronous machines with inertias of the order of $H = 5s$ (Mullane, 2005). The overall network generation rating is approximately 60GVA. This leads to $J \approx 6.1 \times 10^6$ kgm^2 by (2.4). One of the largest potential generation/load imbalance scenarios would be an entire power station of size $\approx$2GW tripping off line. Immediately after the trip, the "missing" 2GW must be supplied to the loads from the inertia of the remaining on-line power stations. Thus, the sum total of energy in their rotors changes with $dE/dt = -2 \times 10^9$ (-2GW) and equation (2.11) reveals df/dt (ROCOF) to be about -0.17 Hz/s. Applying equation (2.11) to different scenarios shows how the expected ROCOF (due to generation-load imbalance) varies dramatically.

Clearly, frequency stability becomes much more of a problem as the size of a power system gets smaller, as reflected in the changes to the expected values of ROCOF. More specifically, the dynamic effects of frequency get much harder to manage as the maximum expected load or generation changes get larger as a proportion of the rotational inertia built into the system.

Scenario	Average per-unit inertia, H (s)	Power system rating	Total system inertia J (kgm^2)	Generation-load imbalance dE/dt	Expected ROCOF (Hz/s)
UK national grid, large load decrease of 100MW	5	60 GVA	6.1×10^6	100 MW	0.01
UK national grid, sudden loss of large power station	5	60 GVA	6.1×10^6	-2 GW	-0.17
1MVA microgrid, removal of 1 10kW load	2.5	1 MVA	51	10 kW	0.1
1MVA microgrid, sudden loss of a 100kVA generator	2.5	1 MVA	51	-100 kW	-1.0
2-generator 100kVA microgrid, removal of 1 kettle (3kW load)	2	100 kVA	4.1	3 kW	0.4
2-generator 100kVA microgrid, sudden loss of one generator	2	100 kVA	4.1	-50 kW	-6

Table 2-1 : Expected ROCOF rates for different power system events

In the case of the UK national grid, even with the loss of a 2GW power station, at -0.17 Hz/s this allows 3 seconds before the normal 1% frequency limit (49.5-50.5 Hz) boundary is crossed (assuming frequency was nominal before the event). For the largest generators in the system, prime mover governors and power outputs will not fully react within this 3 second timeframe (Kundur, 1994), but over 10 seconds most hydro and thermal plants will react according to droop controls, (Kundur, 1994 & Wood, 1996) allowing the frequency excursion to be contained within the 1-1.5Hz bracket, avoiding mass trips in the transmission system and at generation sites. For normal load changes, ROCOF is of the order of 0.01Hz/s. Generator droop controls with relatively slow bandwidths are easily adequate to adjust to such slow changes, while the half-hourly bidding system is a short enough timeframe to efficiently despatch the generators on a unit-by-unit basis.

Contrast this with the situation within a 100 kVA microgrid, where even the addition of a single 13A load causes a ROCOF of about 0.4 Hz/s. In this scenario, the generation systems must fully react within 5 seconds or frequency will drop below 48 Hz. Loss of 50kW of generation within the microgrid at full load would lead to a ROCOF in the region of -6 Hz/s. To avoid frequency dipping below 48Hz this will require immediate despatch of a replacement 50kW of spinning reserve, or immediate shedding of up to 50kW load within 0.3 seconds, or a combination of the two.

2.1.1 Summary of frequency measurement requirements (ROCOF and reaction time)

To allow effective control of microgrid equipment, a frequency measurement algorithm should be able to follow at least 10 Hz/s ramp rates without becoming confused or unlocked. This figure is derived by adding a guard band of 4 Hz/s to the peak value of Table 2-1, which shows that a 6 Hz/s figure could easily be reached within a 100kVA microgrid. Also, the required overall reaction time at 6 Hz/s to remain within the ideal 2Hz window is about 0.3 seconds, of which most will be required for switching or power output changes of prime movers or storage devices. To allow most of the 0.3s for power system output adjustments, the measurement time should be <0.1 second. This implies a measurement in the timeframe of 3-5 cycles.

2.2 Frequency measurement range requirement

BS EN 50160 (BSI, 2000) states that system frequency should always be between 47 and 52 Hz. However, it allows 42.5 to 57.5 Hz in power islands. A frequency measurement should be able to follow a wider range of scenarios. This is because the measurement may be applied at the terminals of an open-circuit machine which may be rotating slowly, or may be over-speeding. Ideally the measurement system would be able to measure from DC to 100 Hz. It must be borne in mind that many measurements will be made via VTs (Voltage Transformers) which have zero gain at DC, so the measurements below 10 Hz may not be possible in practice, although a software algorithm should ideally allow for it. Smaller ranges such as 40-70 Hz might be acceptable so long as the measurement always rails to the correct limit if frequency is outside the measurable range. Fully accurate performance must be achieved over the 40-70Hz range to allow for 50Hz and 60Hz systems. Much lower accuracy is acceptable outside this range since equipment will normally be quickly tripped by under/over-frequency if frequency strays so far from nominal. Also, the accuracy of frequency-dependent controls/algorithms such as synchronising checks and droop controls become irrelevant outside the 40-70Hz range.

2.3 Frequency measurement accuracy requirement

Frequency needs to be measured accurately enough so that all relaying and control functions can be carried out without ambiguity, spurious tripping, oscillation or cyclic control actions. The actual accuracy is made up of 2 factors:-

- absolute accuracy of the clock used within the measuring device. This will affect the absolute error of the measurement, via a DC error bias.

- additional error due to noise, THD, and sampling/algorithm behaviour. This will show up as noise/ripple on the measurement.

Accuracy of cheap crystal oscillators for CPU clocking, taking into account temperature effects, are generally in the range of ±50ppm (±0.005%, equates to a ±0.0025Hz error @50Hz). This magnitude of error is of no consequence for frequency measurement, and can effectively be neglected. Achieving a greater accuracy than this is possible by using oven-stabilised crystals, Rubidium timebases etc., but the expense is not justified in this case, since the additional measurement ripple/noise will dominate the accuracy of the measurement.

BS EN 61000-4-30 (BSI, 2003), which is really a requirement for power quality measuring devices, states that for "class A" performance, the accuracy needs to be ±0.01 Hz (0.02 %). In BS EN 61000-4-30, the expectation is that this measurement is made by counting zero crossings or using a Fourier technique over a ≈10-second sampling window (the window is allowed to be just less than or just greater than 10 seconds, to count an integer number of cycles). Note that a miscount of 1 cycle over 10 seconds would equate to an error of $1/(10*50)=0.2\%$ which is 10x the required accuracy specification. The frequency measurement for control and relaying purposes needs to react within 3-5 cycles (0.1 second), not over 10 seconds. Therefore, meeting a similar ±0.01Hz accuracy specification using only 1/100th of the time window lays down a significant challenge for a measurement system.

Harder upper limits to the measurement accuracy requirement can be estimated by analysing the measurement stability needed to the avoid problems with control, protective relaying and load-shedding algorithms.

For control applications, one estimate of the required frequency accuracy is to imagine a grid-connected generator with 5% frequency droop. Thus, a 5% change in frequency causes a 1pu (1 per-unit) change in output power. Thus, to limit power flow fluctuations as a result of noisy/rippling frequency measurements to less than ±0.01pu, the frequency error/ripple would need to be less than ±0.05%, i.e. ±0.0005 pu (±0.025 Hz for a 50Hz system). This target of ±0.01pu power output fluctuation is set at such a low level in order to both minimise wear on the prime mover (in conjunction with deadbands in the control system), and to reduce the potential for power system oscillations.

Other upper limits are that the frequency error must be <<0.1Hz, to avoid hysteresis/oscillatory problems with load shedding schemes where thresholds are often set in bands approximately 0.2 Hz apart (Moore, 1996b). Obviously, the frequency error should also be much, much, less than the smallest of the under/over-frequency trip limits set by

ER G59/1 (ENA, 1991) (see Table 2-9), which is 1% (0.5 Hz).

Also, if the frequency measurement is to be used to deduce a value of ROCOF for a loss-of-mains (LOM) relay, then the ROCOF might be determined by taking the difference in the output of the frequency algorithm over a 5-cycle timeframe (the proposed latency of the measurement), and dividing by 0.1 (the time which that 5 cycles takes). Thus, if the frequency measurement algorithm has an output ripple/noise of $\pm y$ Hz at any sample, then the ripple on the ROCOF measurement, at 50Hz, will be up to $\pm 2y/0.1$, i.e. $\pm 20y$. Thus, to achieve a ROCOF result with noise/ripple less than ± 0.1 Hz/s, the frequency measurement ripple/noise must be less than ± 0.005 Hz, or $\pm 0.01\%$. This may be an unreachable target in systems with significant harmonic content or instrumentation noise. This is one reason why ROCOF relays have such a poor reputation for spurious tripping, and it is why a different approach to the sensitive subject of LOM relays is proposed later in chapter 6.

Since ROCOF relays include a qualifying time for which the tripping threshold must be exceeded before a trip is registered, it is tempting to carry out a statistical analysis which would allow an increased level of noise on the frequency measurement. This would be done on the basis that the qualifying time requires a number of consecutive samples above a certain threshold, which would reduce the probability of a trip. The problem with this analysis, in the context of microgrids, is that the harmonic content of the waveforms may be high, and the instrumentation sample rate may be low. It will be shown later that in these environments, these effects and constraints can lead to low frequency ripples. These cannot be subjected to a statistical noise-like analysis, and thus the maximum tolerable ripple threshold of ± 0.005 Hz stands for low-frequency error ripples, if the measurement is to be used as the basis for a ROCOF calculation.

In summary, the loosest requirement for frequency measurement accuracy is for a noise/ripple error of $\pm 0.05\%$ (± 0.025 Hz for a 50Hz system), although a small additional DC error term equivalent to the crystal clock accuracy of about $\pm 0.005\%$, (± 0.0025Hz for a 50Hz system) can be tolerated, since it will not introduce any rippling control signals. If a ROCOF calculation is to be made from the frequency measurement, however, then the required error magnitude drops to ± 0.005 Hz, or $\pm 0.01\%$.

2.4 Voltage amplitude measurement speed and accuracy requirements

For measurements of voltage amplitude, the required measurement speed and accuracy depends upon the application. This thesis will propose several different "taps" from a

single initial base measurement, with longer latency measurements having reduced ripple/noise errors. The output from the "taps" can be picked as required for different relaying and control actions.

Similarly to frequency measurements, all voltage magnitude error measurements will be subject to two forms of error:-

- Calibration error; which will be a fixed gain error, plus linearity errors of the VTs and imperfect front-end low-pass filter characteristics for off-nominal frequency inputs. These mechanisms include any interpolation errors between calibration table entries if the calibration tables include points for several off-nominal frequencies or amplitudes to account for filter or VT characteristics. The calibration errors will tend to show up as DC bias offsets on the measurements.
- Additional error due to noise, THD, and sampling/algorithm behaviour. This will generally manifest itself as noise/ripple on the measurement.

An initial hardware calibration might be possible to the 0.1% level using good equipment, and a full on-site closed-loop test conducted carefully. After all error mechanisms such as temperature, linearity etc are accounted for, the total errors due to calibration-related mechanisms could easily account for a 0.5% to 1% error. This error magnitude will appear as a reasonably stable DC bias on the measurement of fundamental by a Fourier technique. Common, economical VTs or voltage measurement transducers are available with ≈1% off-the-shelf accuracy. For practical and economic reasons, on-site closed-loop calibration of a microgrid measurement/control system from VTs to digital sampled data will not be possible. The system will more likely be expected to be simply installed and then operated. Therefore, the overall system calibration errors will be the quoted VT accuracy plus the additional sampling hardware, which can be factory-calibrated. Calibration error for an economical measurement system will thus probably be in the region of 2%.

The fastest voltage-measurement outputs would be required for algorithms within the control systems of power-electronic devices, which are outside the scope of this thesis. Such measurements can be made in <<1 cycle timeframes if the voltage waveforms are assumed to be clean sinusoids, or within a 3-phase system that is balanced.

For under/over-voltage relaying applications, within the microgrid context, the total measurement speed should be less 3 than cycles (60ms), of which some time will be accounted for in the group delay of any anti-aliasing filters. The digital processing latency should thus be less than 2 cycles (40ms). The justification for this statement is presented

in section 2.7.5. Measurement accuracy in this case should be <<0.1pu, since this is the order of magnitude of the steps between tripping thresholds in tables Table 2-8 and Table 2-9. A sensible error would be ±0.02pu, at 1/5 of the smallest (10%) tripping threshold.

For control purposes, a longer timeframe can be allowed, but a reduced noise/ripple level is desirable. A timescale of 5 cycles (100ms) for a measurement is acceptable, considering that AVR control loops and field generators for synchronous generators will not generally react faster than this. To assess an acceptable voltage error level, a grid-connected generator with a 10% reactive droop slope setting is considered. If voltage changes by 10%, the reactive power output control of the generator will change by 1pu. Thus, to keep the reactive power output ripple within ±0.01pu, the voltage measurement ripple/noise must be within ±0.1%, or ±0.001pu. This is a tough target for such a measurement in the context of microgrids and particularly where low sample rates are used. It is, however, a requirement to avoid reactive power ripples which could set up oscillations within a power system. A particular problem with the measurement of amplitude will later be seen to be potentially slow (sub-Hz) oscillations in the measured value of fundamental voltage magnitude, due to aliased harmonics in high-THD environments.

BS EN 61000-4-30 (BSI, 2003), the specification for power quality measurement, specifies an accuracy of 0.1%, 0.001pu, which is measured over a 10-cycle timeframe. This error level is the same as the desired error level deduced above, but the response time is slightly slower than proposed in this thesis, by a factor of 2. Most power quality measurement devices are capable of high sample rates (>=80 samples per cycle), since they are designed to accurately measure harmonics up to the 40th. As mentioned above, the main barrier to be overcome in this thesis, as far as amplitude measurement accuracy is concerned, is to achieve this accuracy with much lower sample rates, down to 10 samples per cycle.

2.5   Current amplitude measurement speed and accuracy requirements

The emphasis of this thesis is on the measurement of voltages. However, all the proposed algorithmic methods are equally applicable to the measurement of currents (and thus to power flows by combining the voltage and current measurements at a node). In terms of measurement speed, overcurrent detection within a graded distribution network protection system does not need to be sub-cycle, but should not take much longer than 1 cycle. The measurement algorithm accuracy requirement for overcurrent detection is quite loose. Therefore, for relaying a single-cycle measurement is probably the most appropriate, and inaccuracy introduced by the digital algorithm will be of no significant

concern unless it becomes greater than that of the CT due to calibration and/or non-linearity. For a protection CT the standard accuracy might be poor at 10% but the linearity good over a range up to 10pu overcurrent (for a 10P10 protection CT) or the accuracy might be better (0.1%) but the linearity poorer under fault conditions (for a Class 0.1 instrumentation CT). (ARW, 2008).

For measurement of power flows, accuracy is more important and measurement times can be longer. A measurement in 5 cycles to match the voltage measurement target is sensible. By far the biggest contributor to the accuracy of the current measurement will be the calibration and linearity of the CT (with associated instrumentation), and how the range of the ADC is set (relative to rated current and potential overcurrent ratio). This is discussed further in section 2.9.

2.6 Phase measurement speed and accuracy requirements

The measurement of phase is not required for fast relaying operations, so a 5-cycle measurement is perfectly acceptable. The phase measurement may be used for one of several purposes (outside of inverter control systems):-

- As a subsection of a frequency measurement algorithm, in which case the phase measurement may be made over 1-cycle and then transformed into a frequency measurement via the rate-of-change-of-phase, before being further filtered

- For the assessment of unbalance via the calculation of negative sequence. To keep unbalance measurements accurate to 0.1% requires the phase measurements from each of the 3 phases to be accurate to ±0.1°. This is because a set of 3 genuine balanced phase voltages with identical magnitudes, but measured relative phases of 0°, -120.1° & -239.9° (i.e. with 0.1° phase errors), results in a calculated unbalance of 0.1%.

- For the assessment of relative phase angles and loss-of-mains (LOM) conditions, where a relative measurement between two points can be made. This requires the two measurements to be made by the same system, by two systems with intimate (low-latency) communication, or by two systems which can timestamp the phase measurements accurately enough that communication latency problems are avoided. This can be achieved, for example, with GPS timestamp information. An acceptable error on such a phase measurement would be of the order of >1° for LOM detection. If used to assess or control power or VAR flow across a transmission line of cable, an accuracy of <1° might be desirable.

- For the calculation of power angles between voltages and currents. These power

angles are required to determine power factor, and the proportions of real and reactive power flowing on each phase. A measured relative phase error of $1°$ between a voltage/current pair, when the power flow was at unity power factor, would result in a perceived reactive power flow of $\sin(1°)=0.017$pu, or a power factor of 0.99985 and is of no concern. The biggest concern with the error in VAR flow measurement would be its effect on a voltage target for a generator in islanded mode (Frequency/Voltage control) with, for example, a 10% voltage droop slope. The resulting voltage target would be shifted by $10\%*0.017=0.0017$pu. This also is of little concern, and would not cause a violation of the flicker limits of Table 2-7 (section 2.7.4) at any ripple frequency. A $1°$ phase measurement error is thus perfectly acceptable for power flow calculations.

Thus, a sensible target accuracy for phase measurements is $±0.1°$, being the requirement to measure unbalance to within ±0.1%.

2.7 Required tolerance to signals with poor power quality

The amplitude, phase and frequency measurements discussed in this thesis must remain robust and accurate under conditions of relatively poor power quality. The expected levels of such disturbances within the UK distribution systems are given in BS EN 50160 (BSI, 2000). An additional useful resource is BS EN 61000-4-30 (BSI, 2003), which describes standard "influence quantities" which measurements must tolerate while still meeting specification, if they are to achieve "class A" accuracy rating.

Within a microgrid, the disturbances such as voltage dips, unbalance, flicker and harmonic content may be significant. The mechanisms for this are described in the following sub-sections. Algorithms to measure amplitude, phase and frequency must be as immune as reasonably possible to these effects, such that the accuracies desired in sections 2.1 to 2.6 can still be met, even during times of worst-case expected interference.

2.7.1 Unbalance

According to BS EN 50160 (BSI, 2000), unbalance should be "within the range 0 to 2%" for "95% of the 10-minute mean RMS values" of unbalance. This does not give a limit on the peak levels of unbalance which may appear for shorter times. BS EN 50160 also states that in some areas, "up to about 3%" may occur.

Within a microgrid scenario, some analysis is required to estimate if much higher figures might be reached. The root cause of increased levels of unbalance will be the increased statistical probability of larger proportional mismatches between the loads on each phase,

coupled with the lower fault levels (higher impedances) within the microgrid. In a large power system, the changing load magnitudes tend to balance on all three phases as there are many thousands or millions of individual loads, split amongst the phases in networks configured by the distribution companies. Also, many of the loads will be balanced three-phase industrial or commercial pieces of equipment. Within a small power system, it is possible that all loads might be single-phase connected, and it is possible that many loads could be active on one phase while far fewer are active on the other two phases. The resulting unbalance can be mitigated by adding 3-phase transformers (Δ-Y$_g$ or Y$_g$-Y$_g$) (Hong, 1997) although this option may be impractical due to cost, weight, size or losses.

In the analysis of unbalance, care must be taken to specify exactly what is meant by the term unbalance. The "true" definition of unbalance, as per BS EN 61000-4-30 (BSI, 2003), is that

$$Unbalance(\%) = 100 \times \frac{NegativeSequenceRMSMagnitude}{PositiveSequenceRMSMagnitude}$$

(2.13)

However, an alternative, given by the IEEE (1991) is

$$Unbalance(\%) = 100 \times \frac{MaximumRMSPhaseDeviationFromAverage}{AveragePhaseRMS}$$

(2.14)

There are key differences between these two formulae. The IEEE definition does not include any phase information (which is required to calculate the negative sequence RMS value), but the IEC definition does not include any zero sequence information. Therefore, it is useful to analyse both the IEC and IEEE unbalance values, and the zero sequence value, in any detailed examination of unbalance.

To determine potential unbalance levels within microgrids, two scenarios are considered:-

a) a 100kVA microgrid which receives its power from a stiff balanced voltage source via a reactance of 0.1pu. This system would be representative of a grid-connected system connected through a delta-star transformer with 0.1pu leakage reactance

b) a 100kVA islanded system connected to a synchronous generator with leakage reactance 0.1pu, via a delta-star transformer, also of leakage reactance 0.1pu. A similar system would be achieved with an inverter connected via an LCL filter which includes a delta-star transformer, with total filter reactance 0.2pu. These are both realistic topologies.

These scenarios were modelled in Simulink, and the results from the islanded cases were, unsurprisingly, found to be the worst cases for unbalance. The model for the islanded case is shown in Figure 2-1.
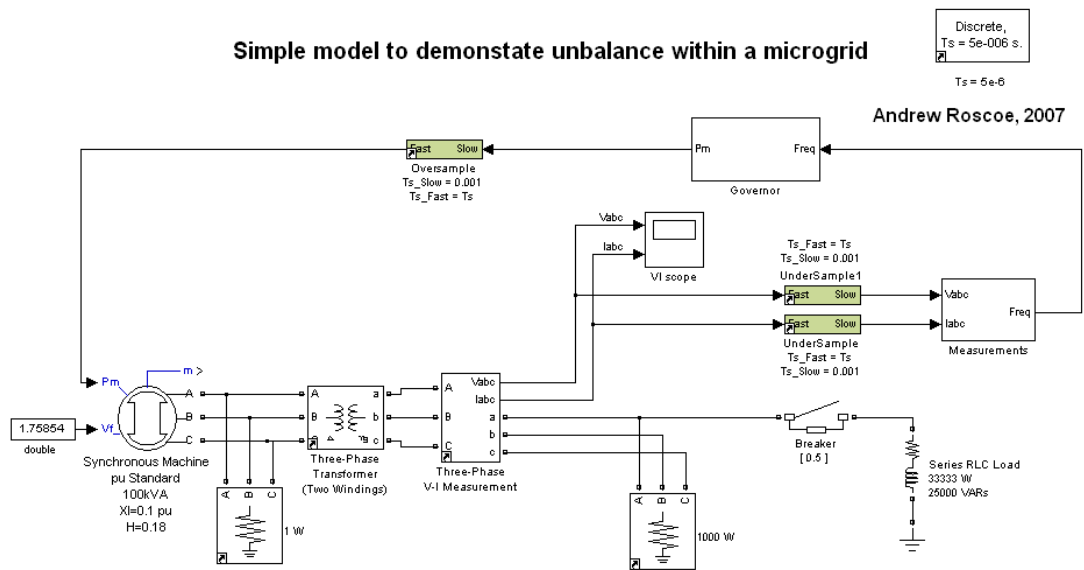
Figure 2-1 : Simulink model to evaluate scenarios of unbalance

The simulations began with very low loading on all three phases (just enough to keep the simulation stable at 5µs step time). During this initial phase, the load was balanced and the phase currents & voltages were also balanced; i.e. the negative sequence and zero sequence components of both current flows and voltages were zero. Then a large single-phase load was added to phase A. The vector magnitudes of the negative sequence and zero sequence components, relative to the positive sequence vector magnitude/phase were recorded, along with the unbalance as defined by the IEEE definition. The results are tabulated below.

Additional load on phase A	10000W	20000W	10000VAR	26667W +20000VAR (33.333kVA PF=0.8)
Normal Unbalance (%) Negative sequence RMS/ positive sequence RMS * 100	3.3 %	6.7 %	3.3 %	**10.2 %**
Negative sequence phase relative to positive sequence phase	-96°	-99°	-178°	-135°
Zero sequence unbalance (%) Zero sequence RMS / Positive sequence RMS * 100	0.6 %	1.1 %	0.6 %	**1.7 %**
Zero sequence phase relative to positive sequence phase	-93°	-95°	180°	-132°
IEEE unbalance (%)	2.2 %	4.5 %	4.3 %	**11.2 %**

Table 2-2 : Worst case unbalance and zero sequence scenarios

Notably, the unbalanced VAR loadings tend to cause a bigger unbalance between the phase RMS voltages, as reflected by the 10000W unbalance causing an IEEE unbalance of only 2.2% while a 10000VAR unbalance causes an IEEE unbalance of 4.3%. The unbalances of real power tend to cause a more subtle phase shifting effect between the phases, rather than a different RMS value on each phase. Unsurprisingly, the worst case is when the single phase is loaded at it's nominal 1pu rating, which is one third of the overall three-phase rating of 100kVA, i.e. 33.333kVA, at a power factor of 0.8, while the two remaining phases are completely unloaded. In this condition, the unbalance reaches >10%, with an additional zero sequence effect of around 2%. This analysis justifies the use of these figures as upper limits on unbalance, during which a measurement device must operate at normal accuracy. The exact magnitudes and phases of the negative and zero sequence components for this worst case are used later in section 2.11 to recreate the unbalance voltage waveforms matching this worst scenario.

A further note on unbalance is that during single or two-phase faults, the system will be up to 100% unbalanced, with a negative sequence component approaching or equal in magnitude to the positive sequence component. These situations may appear only for a short duration, or may persist on a voltage measurement for much longer if, for example, fuses on 2 phases blow but the fuse on the third phase does not. This is a not uncommon situation (as has been experienced in the laboratory at Strathclyde). Any amplitude/phase/frequency measurement must be able to continue operation with sensible outputs during these unbalanced fault events. While even just a single phase measurement is valid, it is a requirement that a frequency measurement algorithm be able to continue operation, thus allowing amplitude/phase measurements on all three phases to also operate correctly. The accuracy of the frequency measurement algorithm may be reduced in this situation, but the algorithm must stay locked.

2.7.2   Harmonic content

The expected harmonic voltage content of LV systems is given by BS EN 50160 (BSI, 2000). The normal allowable levels of each individual harmonic on the voltage waveforms are shown in Figure 2-2. The total allowed voltage THD_V (Total Harmonic Distortion of voltage) should also be <= 8% for 95% of all 10-minute mean RMS values. This means that THD_V may be >8% for 5% of the 10-minute periods, and also that within the 10-minute periods, spot measurements of harmonics and THD_V may be significantly in excess of 8%. Generally, the expected levels of 2n (i.e. even) harmonics are low. The levels of 3n (i.e. 3rd, 6th, 9th etc.) harmonics are also desired to be low, since these set up large circulating currents in the delta windings of transformers (and are attenuated by this effect). This lower tolerance of 3n harmonics shows up most prominently in the specifications for 9th and 15th harmonics

which are well below the allowed values for 7th, 11th, 12th and 17th harmonics. The allowance of 3rd harmonic is also less than that of 5th harmonic. BS EN 50160 does not specify values for harmonics of higher order than 25, although it says they should be "small" and a value of <0.5% is implied.
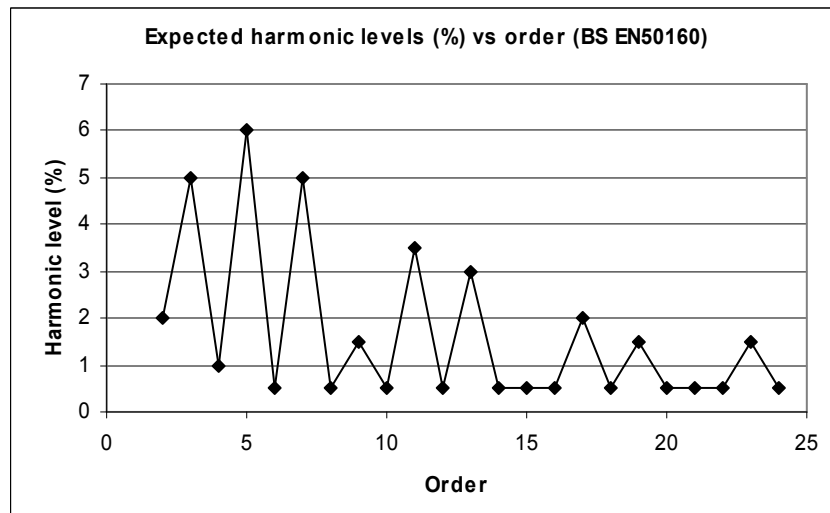
Figure 2-2 : Allowable harmonic levels under BS EN 50160 (BSI, 2000)

BS EN 61000-4-30 (BSI, 2003) expects instruments to meet specification with harmonic levels at "twice the values in IEC 61000-2-4, class 3" (BSI, 2002). Broadly, this equates to harmonics at a level 2.5x that of BS EN 50160, i.e. 2.5x the levels in Figure 2-2.

The chapter on harmonics in (CDA, 2007) suggests that common devices with the worst proportionate harmonic currents are:-

- Older PCs (~ 90% 3rd harmonic, ~70% 5th harmonic, ~50% 7th harmonic, ~30% 9th harmonic)

- Older Fluorescent lights with electronic ballasts (>70% 3rd harmonic, >40% 5th harmonic, ~40% 7th harmonic, ~40% 9th harmonic, ~30% 11th harmonic, ~25% 13th harmonic, ~20% 15th harmonic, ~15% 17th harmonic)

- Motor drives and UPS supplies

Some other devices such as welding equipment are worse, but are less numerous and also in use for shorter (and often sporadic) durations than the above equipment. The above equipment types are common, and are often running for many hours per day.

All these devices may be present within a microgrid scenario. By the list of "worst common devices" above, a valid worst case scenario could be considered as a 100kVA microgrid

feeding large numbers of older PCs and/or fluorescent lights. The microgrid could be supplied by a synchronous generator via a delta-star transformer, of total leakage reactance of 0.2pu (as in section 2.7.1). A relatively simple analysis is to calculate the resulting harmonic voltage levels as the voltage induced across a 0.2pu reactance due to dirty PC/lighting loads drawing a load of 1 pu current at the fundamental. Care must be taken to account for the fact that the source reactance will be larger by a factor of N for each harmonic number N. For example, this means that if the dirty load fundamental current is 1pu, with a harmonic current of 80% at the 3rd harmonic, this will cause a voltage harmonic at 1 x 0.8 x 0.2pu x 3 = 0.48pu, or 48%, due to the harmonic current flowing through the 0.2pu reactance. This will happen if the phases of the third harmonics are all the same; i.e. the load devices are all very similar. This could easily happen in a microgrid feeding PCs of similar brands and model, and similar lighting installations. Diversity in the loads connected would tend to reduce the resulting level of voltage harmonic, as any harmonic currents would become less correlated. Of course, the voltage distortion would also become less if the microgrid was "stiffer" which could be achieved by connection to a stiff parent power network via a transformer of higher rating (and therefore lower leakage reactance), or increasing the local generation capacity on-line.

Using this simple worst-case microgrid scenario, it is possible to estimate absolute worst-case harmonic levels within a microgrid containing only older PCs and fluorescent lights. Such a microgrid might exist if a data-centre was powered from a local generator in islanded mode.

Harmonic	Worst case harmonic current (relative to 1 pu fundamental)	Effective reactance (pu)	Worst case correlated voltage harmonic magnitude
3rd	90%	0.6	54%
5th	70%	1	70%
7th	50%	1.4	70%
9th	40%	1.8	72%
11th	30%	2.2	66%
13th	25%	2.6	65%
15th	20%	3.0	60%
17th	15%	3.4	51%

Table 2-3 : Potential worst case harmonic voltages within a microgrid, older PCs and fluorescent lights

It is evident that these potential levels are much worse than those specified by BS EN 50160. This microgrid would have severe problems due to the magnitude of the harmonic currents flowing and the increased transformer and generator core losses. These pieces of

equipment would have to be over-rated, the loads redesigned, or the proportion of older PCs and fluorescent lights lowered to make the microgrid viable.

Fortunately, the magnitude of this problem is mitigated somewhat by the appearance of BS EN 61000-3-2 (BSI, 2006) and also by the existence of ER G5/4, published by ENA (2005). G5/4 governs the harmonic currents from industrial and commercial equipment, and is specifically designed so that harmonic currents from new installations should not cause BS EN 50160 to be exceeded. BS EN 61000-3-2 addresses domestic and small appliances ≤ 16A per phase, most importantly PCs and fluorescent lighting circuits. The reason that BS EN 61000-3-2 addresses these appliances is because of the reasons shown above in Table 2-3 the large harmonic currents from older devices of these types, and the high probability that many of the same devices are operating for many hours at the same time within localised areas fed from the same transformers/feeders/generators. BS EN 61000-3-2 allows higher limits for appliances like tools, welding kits etc., on the assumption that these pieces of equipment are used less frequently, intermittently, and make up a small proportion of the load. PCs and fluorescent lighting circuits, however, are much more tightly regulated than the data from CDA suggests for older equipment. BS EN 61000-3-2 has been in force since 2001, so a reasonable assumption is that we can use the figures from BS EN 61000-3-2 for new analyses.

BS EN 61000-3-2 gives harmonic current specifications for PCs and fluorescent lights up to the 40th harmonic. The values for PCs are quoted (in BS EN 61000-3-2) in mA/W, which can be approximately converted into percentage harmonic currents. The specifications are shown in Table 2-4. The lighting specifications are much tighter than data from Table 2-3, but the lower order harmonic currents for PCs are still very high[1]. This means that a microgrid consisting entirely of PCs would need to have an over-rated connection to a parent network, or increased generation on-line. However, a reasonable scenario for a microgrid might be to have 50% of the electrical load as PCs, and 50% as lights. Even this limited diversity of loads improves the situation. Data in Table 2-4 is given for up to the 39th harmonic. Although the harmonic currents at the higher harmonic numbers are small, in the region of 3%, these numbers get multiplied by the harmonic number N when converted into harmonic voltages, due to the reactance at the increased harmonic frequency.

_____

[1] Actual measurements of such high levels of harmonic currents from domestic appliances can be seen in Appendix H, "Logged domestic voltage and current waveforms".

Harmonic	Fluorescent lighting harmonic currents (relative to fundamental)	PC harmonic currents (relative to fundamental)
2	2%	
3	30%	78.5%
5	10%	43.9%
7	7%	23.1%
9	5%	11.6%
11	3%	8.1%
13	3%	6.8%
15	3%	5.9%
17	3%	5.2%
19	3%	4.7%
21	3%	4.2%
23	3%	3.9%
25	3%	3.6%
27	3%	3.3%
29	3%	3.1%
31	3%	2.9%
33	3%	2.7%
35	3%	2.5%
37	3%	2.4%
39	3%	2.3%

Table 2-4 : Harmonic currents for fluorescent lights and PCs in BS EN 61000-3-2 (BSI, 2006)

However, it is reasonable to assume that at the 40th harmonic (2000Hz for a 50 Hz system), the harmonic currents summed from all PC and lighting loads are likely to be uncorrelated, even if the devices are similar. Therefore, if there are M separate PCs and lights within the microgrid, the effect of the harmonic currents at the 40th harmonic will be attenuated by $\sqrt{M}$ due to the currents adding in an uncorrelated RMS rather than a coherent superposed manner. Within a 100kVA microgrid with PCs and lights of around 100W each, M will be about 1000, making the $1/\sqrt{M}$ factor 0.031 at the 40th harmonic. A linear taper of this "correlation factor" can be applied between the 2nd to 40th harmonics, with the 2nd harmonics entirely correlated and the 40th harmonics entirely uncorrelated. This linear taper is estimated and not based on hard data. Not including a taper of this kind would, however, result in a vast overestimate of harmonic voltage levels (which will be seen to be very high even with the taper applied), so it is better to include an estimated taper function than not to apply one at all. When this factor is included, and using the example of a 100kVA microgrid with 50% PC loads and 50% lighting loads, the resulting voltage harmonics are shown in Figure 2-3 and Table 2-5.
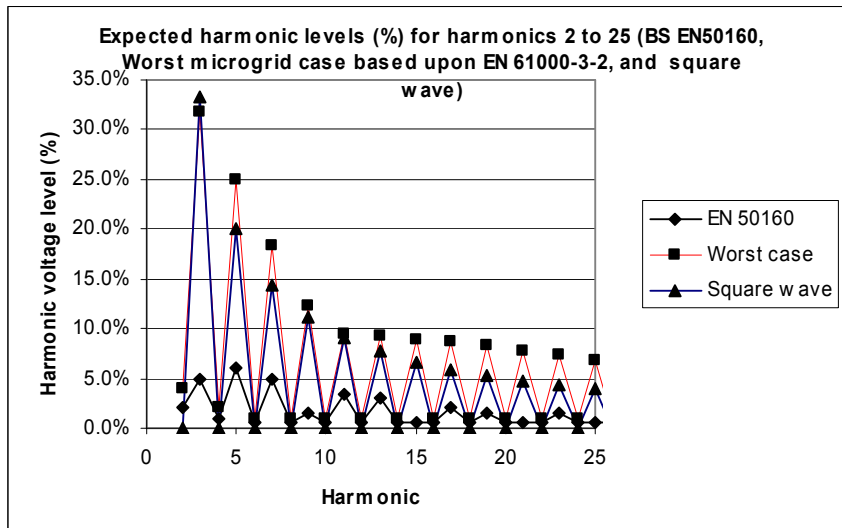
Figure 2-3 : Worst case harmonic voltage levels in a microgrid due to modern PCs and fluorescent lights

	BS EN50160	BS EN50160 After multiplier	Xfmr Effective Reactance	Uncorrelation factor	PC harmonic currents	Lighting harmonic currents	Overall harmonic currents	Overall voltage harmonics	Harmonic level (%) Made at least 2x BS EN 50160 levels
	(%)	(%)	pu		EN 61000-3-2				
Order									
2	2.0%	4.0%	0.4	1.000		2%	1.0%	0.4%	4.0%
3	5.0%	10.0%	0.6	0.975	78.5%	30%	52.9%	31.7%	31.7%
4	1.0%	2.0%	0.8	0.949			0.0%	0.0%	2.0%
5	6.0%	12.0%	1	0.924	43.9%	10%	24.9%	24.9%	24.9%
6	0.5%	1.0%	1.2	0.898			0.0%	0.0%	1.0%
7	5.0%	10.0%	1.4	0.873	23.1%	7%	13.1%	18.4%	18.4%
8	0.5%	1.0%	1.6	0.847			0.0%	0.0%	1.0%
9	1.5%	3.0%	1.8	0.822	11.6%	5%	6.8%	12.2%	12.2%
10	0.5%	1.0%	2	0.796			0.0%	0.0%	1.0%
11	3.5%	7.0%	2.2	0.771	8.1%	3%	4.3%	9.4%	9.4%
12	0.5%	1.0%	2.4	0.745			0.0%	0.0%	1.0%
13	3.0%	6.0%	2.6	0.720	6.8%	3%	3.5%	9.2%	9.2%
14	0.5%	1.0%	2.8	0.694			0.0%	0.0%	1.0%
15	0.5%	1.0%	3	0.669	5.9%	3%	3.0%	9.0%	9.0%
16	0.5%	1.0%	3.2	0.643			0.0%	0.0%	1.0%
17	2.0%	4.0%	3.4	0.618	5.2%	3%	2.5%	8.6%	8.6%
18	0.5%	1.0%	3.6	0.592			0.0%	0.0%	1.0%
19	1.5%	3.0%	3.8	0.567	4.7%	3%	2.2%	8.3%	8.3%
20	0.5%	1.0%	4	0.541			0.0%	0.0%	1.0%
21	0.5%	1.0%	4.2	0.516	4.2%	3%	1.9%	7.8%	7.8%
22	0.5%	1.0%	4.4	0.490			0.0%	0.0%	1.0%
23	1.5%	3.0%	4.6	0.465	3.9%	3%	1.6%	7.3%	7.3%
24	0.5%	1.0%	4.8	0.439			0.0%	0.0%	1.0%
25	0.5%	1.0%	5	0.414	3.6%	3%	1.4%	6.8%	6.8%
26	0.5%	1.0%	5.2	0.388			0.0%	0.0%	1.0%
27	0.5%	1.0%	5.4	0.363	3.3%	3%	1.1%	6.2%	6.2%
28	0.5%	1.0%	5.6	0.337			0.0%	0.0%	1.0%
29	0.5%	1.0%	5.8	0.312	3.1%	3%	0.9%	5.5%	5.5%
30	0.5%	1.0%	6	0.286			0.0%	0.0%	1.0%
31	0.5%	1.0%	6.2	0.261	2.9%	3%	0.8%	4.7%	4.7%
32	0.5%	1.0%	6.4	0.235			0.0%	0.0%	1.0%
33	0.5%	1.0%	6.6	0.210	2.7%	3%	0.6%	3.9%	3.9%
34	0.5%	1.0%	6.8	0.185			0.0%	0.0%	1.0%
35	0.5%	1.0%	7	0.159	2.5%	3%	0.4%	3.1%	3.1%
36	0.5%	1.0%	7.2	0.134			0.0%	0.0%	1.0%
37	0.5%	1.0%	7.4	0.108	2.4%	3%	0.3%	2.2%	2.2%
38	0.5%	1.0%	7.6	0.083			0.0%	0.0%	1.0%
39	0.5%	1.0%	7.8	0.057	2.3%	3%	0.2%	1.2%	1.2%
40	0.5%	1.0%	8	0.032			0.0%	0.0%	1.0%

Table 2-5 : Worst case harmonic voltage levels in a microgrid: 50% fluorescent lights, 50% computers, 0.2pu source impedance

Here, double the value of BS EN 50160 has been used to estimate the worst case even harmonics which are not specified within BS EN 61000-3-2. Notably, the expected level of voltage harmonics is not dissimilar to the levels which would synthesise a square wave. The potential harmonic levels are much worse than BS EN 50160 suggests or allows. If this scenario arose within a conventional UK distribution network, remedial action would be urgently required, either in the form of load removal/redesign, active harmonic cancellation, or stiffening of the grid connection (reduction of impedance).

To illustrate how the harmonic currents in this scenario would cause problems within a grid-connected network, a "K factor analysis" can be performed for the source transformer. The K factor (CDA, 2007) is calculated as

$$K = \sum_{h=2}^{h=h_{max}} h^2 I_h^2$$

(2.15)

where h is the harmonic number and I_h is the proportion of current at harmonic h, relative to the fundamental current magnitude. Performing this analysis on the data in Table 2-4 and Table 2-5, with an average harmonic current formed by a 50% loading of PCs and a 50% loading of lighting, leads to a K factor of 7.0. This means that the approximate core losses in the transformer will be 7 times the core current losses for sinusoidal currents. The transformer must therefore be significantly over-rated to avoid overheating and this is expensive.

The scenario of primary concern within the scope of this thesis is the islanded operation of a local power system containing "dirty loads" (with high harmonic current content). During normal operation, this power system would need to be grid-connected via a relatively low impedance path (i.e. a transformer of high rating) such that the resulting voltage harmonics did not cause violation of the BS EN 50160 limits. This power system might at times be operated in islanded mode using local generation only, for strategic or emergency reasons. The generator/transformer impedance may be significantly higher than the usual grid-connected path impedance. This will result in a lower fault level, and higher harmonic content of the voltage waveforms of the islanded system, by the mechanisms described above. Thus, in this islanded scenario, the harmonic voltage levels of Figure 2-3 and Table 2-5 could in theory arise, if only transiently, and despite the fact that BS EN 50160 would be significantly violated.

To create a consistent test waveform with the above harmonic content, the phases of the harmonics must also be set. Using the standard mathematical formula for synthesising a square wave:-

$$\frac{4}{\pi}\sin(\omega t) + \frac{1}{3}\sin(3\omega t) + \frac{1}{5}\sin(5\omega t) + ...\ \text{(square wave outputting -1 or +1)}$$

(2.16)

and considering that most rectification devices such as power supplies and fluorescent ballasts tend to draw most current at the peak of the cycle, it makes sense to add the odd harmonics with phase offsets of zero such that they add in a fashion which tends to form a sinusoid with the peak clipped. This in the extreme becomes a square wave. This most closely approximates the clipped peaks seen in practice due to such harmonic loads. The even harmonics are considered to have random phases and do not correlate in the same way as the odd harmonics. Applying the worst case expected odd and even harmonics in this manner results in the extremely poor voltage waveform shown in Figure 2-4. The THD_V of this waveform is 53%. Management of harmonic load currents will be important within microgrids to avoid abnormal waveforms such as this. However, the important point here is that measurement algorithms and control software should be able to cope with such poor waveforms, even if they occur transiently for only a few cycles, without becoming confused or giving inaccurate answers.
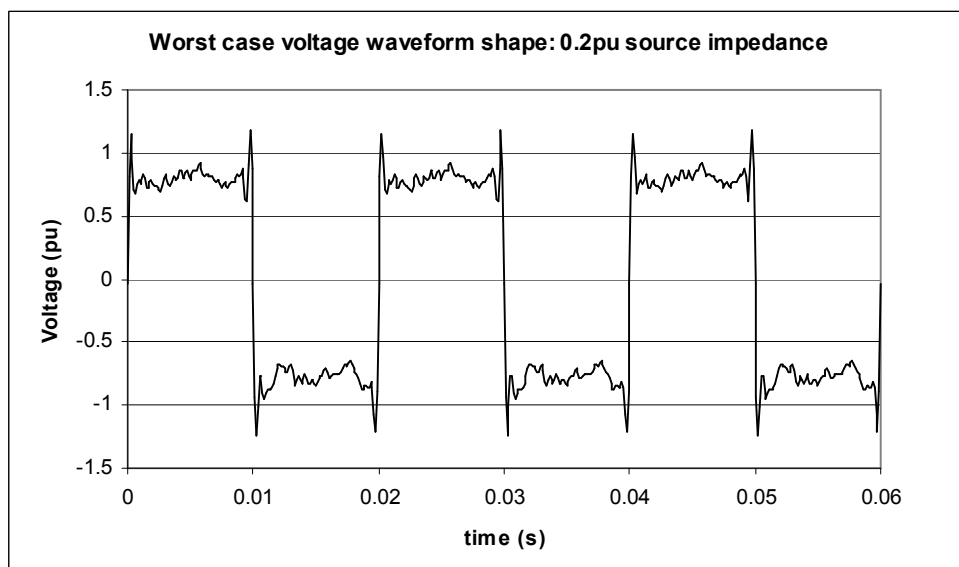
Figure 2-4 : Worst case voltage waveform shape expected within a microgrid: 50% fluorescent lights, 50% computers, 0.2pu source impedance

Although the voltage waveform in Figure 2-4 will be used in this thesis as the worst-case harmonic waveform to test measurement algorithms, its shape is worse than one would hope or expect to see on an AC microgrid for anything more than a few cycles. To demonstrate the way that this poor waveform could be improved in practice, the analysis of section 2.7.2 can be repeated, but using a 0.1pu source impedance instead of a 0.2pu source impedance. In practical terms, this would be achieved by transformer/generator

up-sizing, to reduce either transformer impedance or generator source reactance, or both. The scenario still contains the load made up of 50% fluorescent lights and 50% computers.

The THD_V of this waveform is 28.2%. Compared to Figure 2-4, this waveform is much improved but still highly undesirable.

Order	BS EN50160 (%)	BS EN50160 After multiplier (%)	Xfmr Effective Reactance pu	Uncorrelation factor	PC harmonic currents	Lighting harmonic currents	Overall harmonic currents	Overall voltage harmonics	Harmonic level (%) Made at least 2x BS EN 50160 levels
					EN 61000-3-2				
2	2.0%	4.0%	0.2	1.000		2%	1.0%	0.2%	4.0%
3	5.0%	10.0%	0.3	0.975	78.5%	30%	52.9%	15.9%	15.9%
4	1.0%	2.0%	0.4	0.949			0.0%	0.0%	2.0%
5	6.0%	12.0%	0.5	0.924	43.9%	10%	24.9%	12.4%	12.4%
6	0.5%	1.0%	0.6	0.898			0.0%	0.0%	1.0%
7	5.0%	10.0%	0.7	0.873	23.1%	7%	13.1%	9.2%	10.0%
8	0.5%	1.0%	0.8	0.847			0.0%	0.0%	1.0%
9	1.5%	3.0%	0.9	0.822	11.6%	5%	6.8%	6.1%	6.1%
10	0.5%	1.0%	1	0.796			0.0%	0.0%	1.0%
11	3.5%	7.0%	1.1	0.771	8.1%	3%	4.3%	4.7%	7.0%
12	0.5%	1.0%	1.2	0.745			0.0%	0.0%	1.0%
13	3.0%	6.0%	1.3	0.720	6.8%	3%	3.5%	4.6%	6.0%
14	0.5%	1.0%	1.4	0.694			0.0%	0.0%	1.0%
15	0.5%	1.0%	1.5	0.669	5.9%	3%	3.0%	4.5%	4.5%
16	0.5%	1.0%	1.6	0.643			0.0%	0.0%	1.0%
17	2.0%	4.0%	1.7	0.618	5.2%	3%	2.5%	4.3%	4.3%
18	0.5%	1.0%	1.8	0.592			0.0%	0.0%	1.0%
19	1.5%	3.0%	1.9	0.567	4.7%	3%	2.2%	4.1%	4.1%
20	0.5%	1.0%	2	0.541			0.0%	0.0%	1.0%
21	0.5%	1.0%	2.1	0.516	4.2%	3%	1.9%	3.9%	3.9%
22	0.5%	1.0%	2.2	0.490			0.0%	0.0%	1.0%
23	1.5%	3.0%	2.3	0.465	3.9%	3%	1.6%	3.7%	3.7%
24	0.5%	1.0%	2.4	0.439			0.0%	0.0%	1.0%
25	0.5%	1.0%	2.5	0.414	3.6%	3%	1.4%	3.4%	3.4%
26	0.5%	1.0%	2.6	0.388			0.0%	0.0%	1.0%
27	0.5%	1.0%	2.7	0.363	3.3%	3%	1.1%	3.1%	3.1%
28	0.5%	1.0%	2.8	0.337			0.0%	0.0%	1.0%
29	0.5%	1.0%	2.9	0.312	3.1%	3%	0.9%	2.7%	2.7%
30	0.5%	1.0%	3	0.286			0.0%	0.0%	1.0%
31	0.5%	1.0%	3.1	0.261	2.9%	3%	0.8%	2.4%	2.4%
32	0.5%	1.0%	3.2	0.235			0.0%	0.0%	1.0%
33	0.5%	1.0%	3.3	0.210	2.7%	3%	0.6%	2.0%	2.0%
34	0.5%	1.0%	3.4	0.185			0.0%	0.0%	1.0%
35	0.5%	1.0%	3.5	0.159	2.5%	3%	0.4%	1.5%	1.5%
36	0.5%	1.0%	3.6	0.134			0.0%	0.0%	1.0%
37	0.5%	1.0%	3.7	0.108	2.4%	3%	0.3%	1.1%	1.1%
38	0.5%	1.0%	3.8	0.083			0.0%	0.0%	1.0%
39	0.5%	1.0%	3.9	0.057	2.3%	3%	0.2%	0.6%	1.0%
40	0.5%	1.0%	4	0.032			0.0%	0.0%	1.0%

Table 2-6 : Worst case harmonic voltage levels in a microgrid: 50% fluorescent lights, 50% computers, 0.1pu source impedance
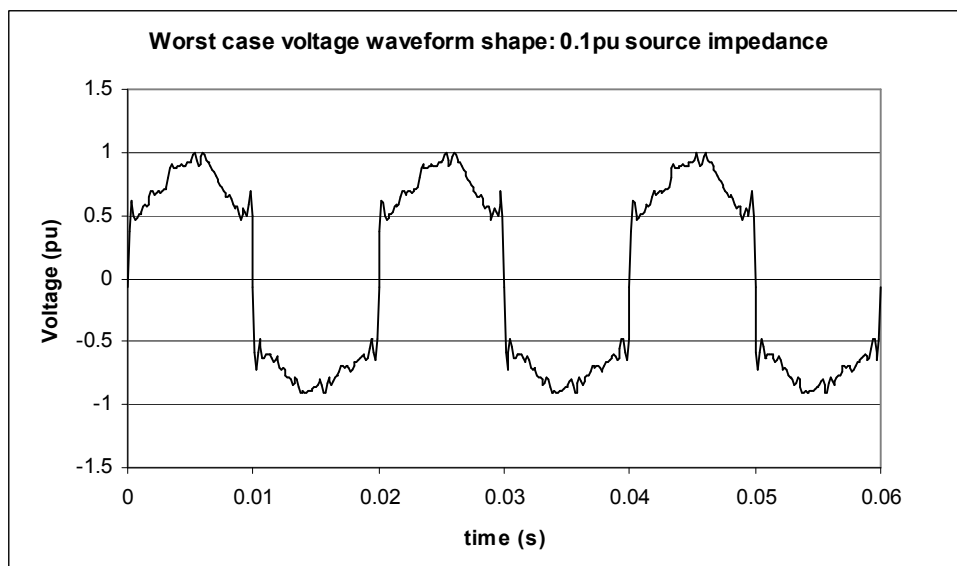
Figure 2-5 : Worst case voltage waveform shape expected within a microgrid: 50% fluorescent lights, 50% computers, 0.1pu source impedance

2.7.3   Inter-harmonic content

Inter-harmonics are caused from two main sources.

The first source is devices which switch at frequencies unrelated to the fundamental frequency. An example would be an inverter-connected generator. The switching transistors or IGBTs are connected to the power system via an LC or LCL filter which smoothes out the fast switching pulses to achieve an approximately smooth sinusoidal voltage and current waveform. Typically, the primary inductor has a reactance of ≈0.15pu, and the LC filter cut-off frequency is set at around 500Hz for a sensible switching frequency which typically may be in the range 2-20kHz. The worst case voltage inter-harmonic results when the inverter is feeding an islanded power system, and the inverter is lightly loaded or open circuit, as the switching pulses are smoothed only by the LC filter and not by an LC-LR filter which is formed if a load R is added. The voltage gain of the LC filter is given by {B.4}. The worst case attenuation of a 2025Hz switching frequency with a 500Hz resonant LC frequency is therefore about 15 (24dB), leading to an inter-harmonic at 2025Hz of 6.5%. BS EN 50160 does not regulate these inter-harmonics as yet, although allowable levels are "under consideration, pending more experience". It does, however, correctly note that some inter-harmonics can be regarded as a form of flicker (see section 2.7.4).

Other inter-harmonics can be caused by devices which communicate via power lines. BS EN 50160 limits these signals to the levels shown in Figure 2-6, over 99% of a day, measured as

the three-second mean signal voltage levels. Note that this means the signals may transiently be higher than this. Also, multiple interfering signals may be present simultaneously with components at several frequencies, although Figure 2-6 gives no indication of the spectral density expected. This is a serious ambiguity in the BS EN 50160 specifications. A reasonable way of simulating such signals is to create a square wave at a sensible signalling bit rate. The square wave will introduce a spread of signals by equation (2.16). Since the maximum fundamental component of the signalling spectrum at the signalling bit rate might have a temporary maximum level twice that of Figure 2-6, then the square wave magnitude of the signalling voltage might be 2*9% x 4/π = 23% relative to the nominal voltage magnitude, at a frequency of 525Hz. This is a large interfering signal! It is hard to believe that such a signal would a) be acceptable and b) possibly be injected, due to the large power levels required from the signalling device. However BS EN 50160, as it stands, allows signals of this magnitude to be present on mains voltages.
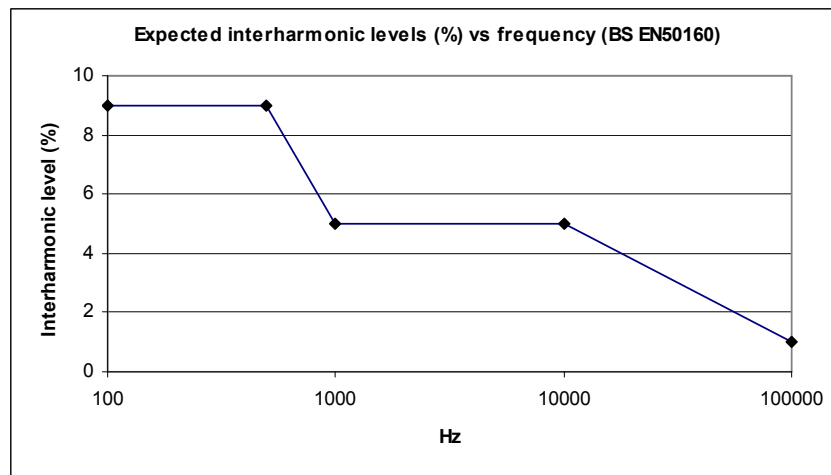
Figure 2-6 : Expected average inter-harmonic levels due to signalling, from BS EN 50160

In a worst-case voltage waveform, used to test measurement algorithms, the following interfering signals should therefore be included:-

- A 6.5% sinusoidal inter-harmonic at 2025Hz (to simulate inverter generation or local customer communication)

- A 23% square wave at 525Hz, to simulate utility power-line communication

2.7.4   Flicker

Flicker is a modulation of the voltage sine wave envelope at frequencies from <1Hz to >1000Hz. However at frequencies >25Hz the flicker can be regarded instead as an inter-harmonic, and at frequencies <<1Hz the flicker is not really repetitive but more a succession of steps in voltage. The analysis here focuses on the 0.1Hz to 25Hz band. The

54

primary concern of flicker is the "annoyance" it causes to people due to the varying brightness of incandescent light bulbs at frequencies within this band. The method of flicker measurement is relatively complicated, and is defined in BS EN 61000-4-15 (BSI, 1998). The measurement is designed to output numbers for P_{st} and P_{lt}, short and long-term average values, which represent the perceived annoyance caused by a flickering lamp. The allowed values of P_{st} and P_{lt} are laid down in BS EN 50160. P_{st} is measured over 10 minutes, and P_{lt} is an average of 12 consecutive P_{st} values over 2 hours. P_{lt} must be ≤1 for 95% of the time.

P_{st} and P_{lt} are determined statistically from the "real-time flicker sensation" output of a multi-stage filter, which includes a band-pass filter centred on 8.8Hz which is the most annoying flicker frequency to the human eye/brain (BSI, 1998). The filters were originally purely analogue devices, and digital implementations need to be coded carefully to accurately match the performance of the analogue equivalents. The statistical analysis is based upon empirical analysis and experiments with people subjected to different frequencies and depths of flicker.

The "real-time flicker sensation" output from block 4 of the flicker meter (BSI, 1998) is converted into P_{st} by the following formula:-

$$P_{st} = \sqrt{0.0314P_{0.1} + 0.0525P_1 + 0.0657P_3 + 0.28P_{10} + 0.08P_{50}}$$

(2.17)

where $P_{0.1}$, P_1, P_3, P_{10}, P_{50} are the real-time flicker levels exceeded for 0.1%, 1%, 3%, 10% and 50% of the time within a 10-minute window. Thus, a constant flicker level of 1.962 will lead to P_{st}=1, which is the BS EN 50160 limit. However, flicker levels of just below 1/0.0314=31.85 which only occur for 0.1% or less of the 10 minute window (0.6 seconds) may not cause P_{st} to rise above 1 if the flicker level is very low for the remaining 99.9% of the time.

BS EN 61000-4-15 gives example values of steady sinusoidal and square-wave modulation depths and frequencies which cause P_{st} to equal 1. These are given in Table 2-7.

Analysis of the flicker meter specified by BS EN 61000-4-15 reveals that the real-time flicker sensation is proportional to the step magnitude squared. Thus, at 1620 steps per minute (13.5 Hz modulation), a step magnitude of 0.402% x $\sqrt{(31.85/1.962)}$ = 1.62% will give rise to a real-time flicker level of 31.85. A short-term flicker level this high for only 0.6 seconds will cause P_{st} calculated over 10 minutes to be 1, the BS EN 50160 limit. This

could be regarded as a worst-case acceptable flicker magnitude and frequency within a normal UK grid-connected system.

Voltage changes per minute	Step magnitude (% of nominal)
1	2.724%
2	2.211%
7	1.459%
39	0.906%
110 (0.9 Hz modulation)	0.725%
1620 (13.5 Hz modulation)	0.402%
4000	2.4%

Table 2-7 : Voltage step changes to cause Pst=1 in a 50Hz system

An alternative viewpoint is to examine microgrids with a weak network connection and a relatively large amount of installed wind generation. The inductive impedance of the network connection is not of prime importance, as we imagine that the wind turbines design allows them to operate at unity power factor. However, gusty wind will inevitably lead to fluctuating real power flows across this boundary, and a resulting fluctuating voltage magnitude due to the resistance of the network connection. A simple scenario is for a rural farm connected via an 11kV overhead line to a stiff grid. The maximum capacity of this line is >4MVA (see appendix C.1). Assuming that the actual connected load is 500kVA or less, then by equation {B.3}, the maximum line length could be 24km (setting k to 0.1 in {B.3}), which is a realistic line length in some rural areas. If the farm has 100kW of installed wind capacity, then gusty wind could cause a 100kW fluctuation in real power flow across this 24km line, equating to 5A RMS per phase. The resistance of the 24km line would be about 15Ω (by Tab. C-1). The resulting voltage drop between a single phase and neutral of the 11kV system at the farm connection would be 5A x 15Ω = 75V RMS. Relating this to the nominal phase-neutral value of $11000/\sqrt{3}$ gives a 1.2% fluctuation.

A second scenario is for a similar 100kW fluctuation to occur in a 400V cable-connected microgrid of capacity up to 200kVA. If the cable connection is 500m long, with 0.25Ω/km, then the 100kW power fluctuation causes a 144A RMS fluctuation, which leads to an 18V RMS drop in phase voltage. Expressed at a percentage of nominal 231V, this is almost 8%. Thus, one source of bad flicker may be encountered when the LV connection length is long, and there are substantial fluctuating power flows due intermittent generation or loads.

A reasonable limit on flicker step size is therefore 8%, partly by the analysis above and partly because larger steps will risk tripping due to under-overvoltage. It is unlikely that such steps will occur at the worst frequency for flicker (8.8Hz), but BS EN 61000-4-30 specifies that power quality meters must operate within specification with flicker levels of P_{st} up to 20. To achieve P_{st}=20 requires equation (2.17) to give 20. Reversing (2.17) leads to a constant level of real-time flicker of 785. At 13.5Hz, the constant repeating step magnitude to achieve a real-time flicker of 785 is 0.402% x $\sqrt{(785/1.962)}$ = 8%! A square wave modulation at 13.5Hz of size 8% is an extremely unpleasant prospect, and, similarly to the worst-case harmonic and inter-harmonic levels deduced in sections 2.7.2 and 2.7.3, will hopefully never be encountered for any length of time. The fact remains, however, that such effects may appear, if only for a few cycles. Since the measurements addressed in this thesis are made on the same timescales of a few cycles, the measurements must be able to cope with these effects.

2.7.5   Tolerance to voltage dips and surges

Voltage dips and surges occur on power networks due to switching, faults, and other disturbances.

Electrical loads are generally designed to withstand voltage dips and surges of certain magnitudes and durations. There are several different design curves which have evolved over time (CDA, 2007), of which three are:-

- the CBEMA (Computer and Business Equipment Manufacturers Association) curve

- the ITIC (Information Technology Industry Council) curve.

- the ANSI IEEE 446

The curves are all similar. A copy of the CBEMA curve is shown in appendix C.2. Equipment is usually designed to operate without malfunction for the voltage dips and surges within the upper and lower bounds. Useful example points are that a 100% voltage dip should be survivable for 20ms, and that a 20% dip should be survivable for about 1 second. Of course, dips and surges of ≤±10% must be survivable indefinitely since this is the steady-state voltage range specification. BS EN 50160 gives some more anecdotal information, which is that most dips have durations of <1 second and a depth of <60% (i.e. dips to ≥0.4pu). In some areas where grids are weak, 10-15% dips can occur frequently.

A useful reference is IEEE 1547 (IEEE, 2003) which describes the current standard for interconnecting distributed generation to electric power systems in the US. The guidelines and required trip times given in this document are pertinent to microgrid applications. In

the future the requirements may become looser or tighter as microgrid and distribution network control technology advances, to account for higher penetrations of distributed generation. Table 1 of IEEE 1547 lists the required total clearing times (from the start of the abnormal condition to actual disconnection), for various voltage dips and surges. This table is recreated here:-

Voltage range (% of base voltage)	Maximum clearing time(s)
V<50	0.16
50≤V<88	2.00
110<V<120	1.00
V≥120	0.16

Table 2-8 : Under/Overvoltage clearing times required under IEEE 1547

A similar table is also contained within ER G59/1 (ENA, 1991), the guidelines for connecting distributed generation in the UK, up to 5MW or 20kV:-

Protection	Phases	Trip setting	Maximum clearing time(s)
Under-voltage	All	-10% (phase-neutral)	0.5
Over-voltage	All	+10% (phase-neutral)	0.5
Under-frequency	1	-6%	0.5
Over-frequency	1	+1%	0.5

Table 2-9 : Protective equipment and settings for LV supply arrangements

The required tripping times define the maximum allowable reaction time for voltage amplitude measurements which are used for over/undervoltage relaying. To meet the required tripping time, the reaction time of the measurement & digital processing plus the reaction time of the breaker must be less than the shortest time on Table 2-8 (160ms). Accounting for a time of between 40ms for an air-blast breaker (Laughton, 2003) and 100ms for an oil-filled breaker (Areva T&D, 2007) for contacts to open and arcs to be extinguished, then approximately 60ms still remains available for the latency of the measurement and processing. This is 3 cycles, of which some time will be lost due to the group delay within anti-aliasing filters etc. Therefore, the requirement for a <2 cycle digital processing time quoted in section 2.4 is justified. Provided the amplitude/phase measurements have a valid estimate of frequency, voltage measurement algorithms for under-voltage/over-voltage relaying activities should respond accurately within this timeframe.

In terms of frequency measurement, however, all the data above needs to be regarded in

a different way. Now, the requirement is that a frequency measurement algorithm must continue to give a sensible output during such events. The events may include a full 3-phase fault with all three phase voltages at 0pu. For allowable lengths of time, the local breaker should not be tripped, to allow local equipment which can ride through a brownout, to do so. If the frequency measurement algorithm cannot ride through the event, then one of a number of things may occur:-

- A spurious under/over-frequency trip

- A spurious under/over-voltage trip, due to amplitude/phase measurement algorithms being given the wrong value of frequency

- A spurious LOM (loss of mains) trip, if the loss-of-mains protective algorithm uses any combination of the frequency, amplitude or phase measurements, on any combination of phases.

So, if a full 3-phase to ground fault or momentary disconnection occurs, the frequency measurement algorithm must be able to hold its output at some last known "good" value, for a configurable time, before being forced to revert to an actual measured value. This action can be called "ride-through" and is highly desirable (Moore, 1996a & 1996b). The time limit should be configurable since the sources of information above give conflicting advice on how long this time should actually be. It will thus be an application-specific parameter. This time length will almost certainly be ≥20ms, since the CBEMA curve suggests that local equipment is designed to ride through this length of brownout. The time length will probably be less than 0.5s, which is the longest tripping time for a 100% dip, specified by ER G59/1. IEEE 1574 is in the middle, with a figure of 160ms.

For dips which occur on only 1 or 2 phases, the requirement must be that the frequency measurement should continue to operate indefinitely with reasonable accuracy, as described in the section on unbalance (section 2.7.1)

Suitable requirements upon the frequency measurement algorithm for balanced (three-phase) dips and surges are thus:-

- maintain standard accuracy during 20% dips

- maintain reasonable accuracy during 60% dips up to 100ms (voltage at 40% of nominal)

- output a sensible value during 100% dips for 20ms using ride-through code (this time configurable within the algorithm so that it can be changed "in the field")

- recover to reasonable accuracy within 3-5 cycles after a dip, whenever voltage rises to 40% of nominal or above

- recover to standard accuracy within 3-5 cycles after a dip, whenever voltage rises to 80% of nominal or above

2.8   Loss of mains requirements

The requirements on frequency, amplitude and phase measurements for achieving adequate loss-of-mains detection are not immediately clear. Both G59 and IEEE 1547 are extremely vague on the acceptable algorithms and thresholds for such relaying, and the only definite information is that IEEE 1547 specifies a detection time limit of 2 seconds. Therefore, for the purposes of amplitude, frequency and phase measurement, a sensible approach is to make the measurements meet all the other requirements as well as possible, and then see how well the measurements can be applied to loss-of-mains detection. This topic is addressed in detail in chapter 6.

2.9   Measurement hardware and sample rate considerations

The measurement algorithms must be able to be made on hardware which is relatively cheap, small, and integrated with other measurement and control functions. This is because the target market for the algorithms is distributed generators and small microgrid power systems. As such, the emphasis is on many, cheap installations rather than few expensive ones. The cost of the measurement hardware and processor must be kept low so that it does not become a significant part of the system cost. Because of this, it is desirable to combine the measurement algorithms developed in this thesis with all the other local microgrid control algorithms into a single piece of code that operates on a single microcontroller at a fixed frame rate. This places limits on the frame rate since the overall process may contain much code. Other constraints on frame rate may be processing overheads such as data logging and communication with external devices. Experience at Strathclyde with microcontrollers shows that data logging in particular can be a severe constraint on frame rate, due to the access speeds of suitable memory areas. This is true even when logging decimation is high (i.e. logged data is only captured once every X frames), since the limit on frame rate is set by the longest potential frame time and not the average frame time.

Another constraint on the algorithms is that, in general, it will not be possible to synchronise the ADC sample points with zero crossings on the measured waveforms. Some specialised digital relays, and some power quality meters, use specialised sampling and CPU hardware which operates at variable frame rates (as described for example by Moore

(1996a)). The frame rate and sample timing can be determined by a PLL locked to the system frequency. This allows synchronisation of the zero crossings with sample times, and allows the number of samples per cycle to be kept constant. The hardware is specialised and costly, and also the PLL dynamics affect the dynamic performance of the system in a way which may not be appropriate in a microgrid with high rates of change of frequency as described in Table 2-1. Within the remit of this thesis, this approach is not available.

To make the algorithms function on more widely available (and cheaper) processing platforms, the ADC samples will therefore be taken at a fixed frame rate, synchronised to the fixed CPU frame rate. An achievable frame rate, based upon experience of integrated microgrid control algorithms at the University of Strathclyde, is 10 Sa/cycle or 500 Sa/s. This equates to a frame rate of 2ms, and allows relatively large microgrid control code algorithms to be executed successfully on realistic microcontroller systems, taking into account data logging and communication requirements.

The accuracy and quality of any measurements is affected by many mechanisms within the chain of hardware and software which forms the power system, instrumentation, and processing. In summary, these errors and mechanisms are:-

- Actual noise, spikes and harmonics present on the power system voltages and currents.

- The accuracy of voltage and current transducers (VTs, CTs, plus their burdens, or other measurement devices such as optical or Hall-effect sensors). Amplitude accuaracy, phase accuracy/lag and linearity are all defined by the design of the transducers.

- Noise, interference and cross-talk in cables.

- Amplitude accuracy, phase accuracy/lag, linearity and noise in any instrumentation/isolation amplifiers and anti-aliasing filters.

- Amplitude accuracy, timing skews/jitter and bit noise of ADC measurements.

- Errors due to sample rate (interpolation), mathematical approximations, and aliased harmonics within the digital processing system (mathematical algorithms).

To estimate the signal/noise ratio of relatively cheap measurement hardware, measurements were taken of the voltage measurement instrumentation circuits at the University of Strathclyde. The instrumentation consists of three-phase 400V/110V star-star VTs, with each phase connected via lengthy, shielded, treble twisted-pair cables through an electrically noisy environment to a set of isolation amplifiers. These are based upon the

ISO124P isolator, with some additional components including an amplifier/filter stage with a basic 741 op-amp and a 3kHz low-pass filter which attenuates the 500kHz modulation used within the ISO124P. The Gaussian and 500kHz noise from these amplifiers after filtering is approximately 20mV RMS, with the 1pu (peak) signal amplitude set to ±5V. This equates to a noise level of approximately 0.005pu RMS, or a signal-to-noise ratio of 200 (46dB) which is relatively poor, and presents a sensible worst value to an expected hardware noise specification. It could almost certainly be improved using circuits designed commercially, and/or by lowering the cut-off frequency of the low-pass filter. A worst case noise level (for a voltage measurement) is therefore 0.5% of the nominal measured signal level. Note that for current measurements, the noise level may be significantly higher. This is due to:-

- The ADC range which may need to encompass much larger over-ranges than ±2pu to be able to measure fault currents

- The current flowing in a system will often be at a level <<1pu, when loads or generation levels are at only a fraction of the branch capacity.

A conventional ADC resolution is 12 bits. A sensible signal scaling (for a voltage input) is that a 1pu peak-peak input signal voltage spans half the range of the ADCs, allowing for linear measurements up to ±2pu, but with higher signals clipped to the 2pu peak positive or negative values. In this work, ADC non-linearity effects are ignored, the justification being that the harmonic distortion content of the expected signals (possibly >>8% THD) is far larger than any reasonable ADC non-linearity specification. With many practical ADC setups, the lowest bits of the ADC can become unusable depending upon the hardware, software application, and the care with which it is set up. Therefore, a sensible precaution is to allow for an additional RMS quantisation noise of an RMS magnitude equal to 2x the LSB (least significant bit). This effectively scrambles the 2 least significant bits and means that only the top 10 bits are really usable.

It is contextually useful here to tabulate and compare the effective RMS noise levels from actual noise and from ADC quantisation noise. Here, the RMS noise due to a quantisation step of size a is given by $\sqrt{(a^2/12)}$ which is derived from the standard formula for the standard deviation of a uniform distribution. A further, easy to derive formula is that

$$\frac{x}{S} = N.2^b.\sqrt{3}$$

(2.18)

where the ADC input is scaled over the range -x to +x pu, S is the actual per-unit input signal input, there are b usable ADC bits, and N is the noise equivalent per-unit RMS value

(relative to signal level S). Some possible scenarios are:-

RMS noise (pu)	Equivalent quantisation noise	Potential scenario
0.000282	1pu input, 12 bits over -2 to +2 range	Best possible from a 12-bit ADC
0.00113	1pu input, 10 usable bits over -2 to +2 range	Unusable (noisy) 2 LSBs
0.005	1pu input, 12 bits scaled over -35 to +35 pu range OR 0.28pu input from a CT, 12 bits scaled over -10 to +10 pu range for overcurrent measurement.	Measurement of 1pu current in setups with wide range inputs, OR, Measurement of small current flows.

Table 2-10 : Equivalent noise contributions of ADC quantisation effects

The effective RMS noise with a 12-bit ADC will therefore never be less than 0.000282pu. It may be as high as 0.005pu (-46dB) for measurements on poorly instrumented systems (Table 2-10). For current measurements, the RMS noise may be significantly higher on a per-unit basis, where low levels of current are measured and/or the ADC range is configured to measure wide ranges of over-current.

2.10 Overall amplitude and frequency measurement specifications (for voltage measurements)

The previous sections can be summarised into the requirement specifications for measurements of 3-phase voltage amplitude, phase and frequency. These requirements apply to measurements of these dynamic parameters within microgrid scenarios. In these scenarios, ROCOF rates are potentially high, power quality is potentially very poor, and response times must be appropriate for the required trip times and control dynamics. For relaying actions, accuracy must be appropriate to avoid spurious trips and missed trips. For control actions, measurements must contain very low levels of ripple to avoid passing this ripple back (potentially amplified due to droop controls) to prime mover or generator controls.

Description	Target
Absolute error	±0.02pu pk
Ripple and noise (ignoring DC biased in the absolute error)	±0.02pu pk
Response time (latency)	2 cycles
Measurable range	0.01 to 2pu

Table 2-11 : Voltage amplitude measurement specifications (relaying)

Description	Target
Absolute error	±0.02pu pk
Ripple and noise (ignoring DC biased in the absolute error)	±0.001pu pk (may not be achievable at the lowest sample rates with high THD levels)
Response time (latency)	5 cycles
Measurable range	0.01 to 2pu

Table 2-12 : Voltage amplitude measurement specifications (control)

Description	Target
Absolute error	±0.1° (probably only achievable with closed-loop calibration of the entire measurement system)
Ripple and noise (ignoring DC biased in the absolute error)	±0.1°
Response time (latency)	5 cycles
Measurable range	0.8 pu to 2pu voltage magnitude

Table 2-13 : Phase measurement specifications (control/instrumentation)

Description	Target
Absolute error	±0.025Hz pk
Ripple and noise at steady state, 1pu on all phases, at or near nominal frequency (ignoring DC bias in the absolute error)	±0.025Hz pk (±0.0005 pu) for standard uses ±0.005Hz pk for ROCOF relays (may not be achievable at the lowest sample rates with high THD levels)
Response time (latency)	5 cycles (0.1 seconds)
Measurable range	40 ≤ Freq ≤ 70 with full accuracy. Also 30 ≤ Freq ≤ 80 with reduced accuracy of ±0.25Hz (to cope with underspeed/overspeed). Also 10 ≤ Freq ≤ 100 with further reduced accuracy of ±0.5Hz (to cope with underspeed/overspeed). Must not measure a sub-harmonic or harmonic. Must rail to the correct upper or lower limit if frequency is outside the measurable range.
Ride-through capability	For a configurable time, during 3-phase dips to less than 0.05pu, a ride-through action must hold the last known "good" frequency measurement, until the configurable timer elapses or the dip finishes, whichever occurs first.

	Maintain standard accuracy during 20% dips
	Maintain slightly reduced accuracy during single or two-phase faults. Ideally to ±0.010Hz for single-phase faults and ±0.015Hz for two-phase faults.
	Maintain reasonable accuracy of ±0.25Hz during three-phase dips up to 60% (voltage at 40%-80% of nominal)
Allowance for reduced accuracy during sustained low voltage events	Maintain reasonable accuracy of ±0.5Hz during three-phase dips up to 95% (voltage at 5%-40% of nominal)
	Allow an extra ripple/noise on the measurement equal to the steady-state specification, for each phase dropped below 5% of nominal voltage.
	Recover to standard accuracy within 5 cycles after a dip, whenever voltage rises to 80% of nominal or above.

Table 2-14 : Frequency measurement specifications

Description	Target
Sample rate	500 Sa/s (nominally 10 Sa/cycle) if possible
Maximum system ROCOF	10Hz/s
Unbalance	10% negative sequence, plus 2% zero sequence
Harmonic distortion tolerable wile meeting steady state accuracy	Harmonics as described in Figure 2-3 and Figure 2-4, section 2.7.2. THD 53%
Inter-harmonic distortion	6.5% sinusoidal inter-harmonic at 2025Hz plus 23% square wave at 525Hz
Flicker	8% step flicker at 13.5Hz
Sudden phase jumps due to switching of loads	10 degrees, which would be experienced by removal of a 1pu power flow across a 17% pu reactance transformer/transmission line combination. NOTE: Frequency measurement output will transiently be in error subsequent to such a disturbance.
ADC quantisation noise	12-bit ADC scaled so that nominal input signal at 1pu spans half the ADC range, with clipping to the 2pu +ve and –ve peak signal levels.12, with 2 bits additional RMS ADC sampling noise
Gaussian noise level (RMS) due to pre-ADC instrumentation and filtering	0.5% pu RMS (46dB SNR)

Table 2-15 : Range of interfering influence qualities and constraints

2.11   Generation of suitable test waveforms.

To simplify testing, four waveforms of length 60 seconds have been generated, against which candidate algorithms can be tested. The waveforms are designed to test the criteria of Table 2-11 to Table 2-14, under the interfering influences described in Table 2-15. The signal distortions of Table 2-15 are switched on and off so that the effects of all the influences upon a given algorithm performance can be analysed from a single simulation.

Waveform 1 contains some extremely dynamic ROCOF events up to ±10 Hz/s, plus a 10° phase jump, and also tests the full frequency range from DC to 100Hz. It is the main test waveform for the frequency measurement algorithms (see section 5.7).

Waveform 2 tests frequency and amplitude measurements with a much lower level of ROCOF (0.2Hz/s) and over a more restricted range of 44 to 55 Hz. This waveform is useful for verifying that there are no particular problems at particular frequencies.

Waveforms 1B and 2B contain reduced levels of THD_V and instrumentation noise, and also have no flicker applied. These are used to verify the performance of the amplitude measurements in chapter 4.6. Waveform 1B is similar to Waveform 1, but with the following changes:-

- Flicker is not applied. This is because the flicker is (quite correctly) picked up by the amplitude measurement, making assessment of the steady-state accuracy difficult under conditions of large flicker step magnitudes.

  THD_V is reduced from 53% to 28%, as per

- Table 2-6. The instrumentation noise is reduced from 0.005pu RMS to 0.001pu RMS (60dB SNR), and the additional ADC quantisation noise is reduced from 2 bits RMS to 1 bit RMS. As will be shown in chapter 3, section 4.7, meeting the ±0.001pu amplitude measurement ripple specification is only possible for signals with this level of THD (or less), and by achieving improved instrumentation noise levels. The amplitude measurements are still robust and stable under the conditions of 53% THD_V and 46dB SNR, but the accuracy/ripple at steady state is about ±0.003pu.

Waveform 2B is similar to Waveform 2 in the same way that Waveform 1B is similar to Waveform 1; with the same modifications to flicker, THD and instrumentation noise.

2.11.1   Waveform 1

Time	Description	Purpose / Test
0-1	NOTHING (instrumentation/ADC noise only)	Simulate no connection to network
1-2	42 Hz, 1pu on all phases	Deliberate off-nominal frequency.
2-3	NOTHING (instrumentation/ADC noise only)	Simulate disconnection from network
4-8	51.282 Hz (9.75 samples/cycle), with instrumentation/ADC noise	Test settling time to a new frequency
8-9	with unbalance added	Adds ripple to 3-phase measurements
9-9.5	with harmonics added	Disturbs all algorithms
9.5-10	with inter-harmonics added	Simulate mains signalling
10-10.5	with flicker added	Simulate worst case local flicker sources
10.5	phase jump 10 degrees	Simulate network switching
11	begin a +1 Hz/s ramp until 14 seconds	Dynamic frequency for following faults
11-11.04	3-phase dip 100%	Ride-through ability
11.5-11.75	3-phase dip 95%	Very low signal levels (low SNR)
12-12.25	3-phase dip 60%	Intermediate signal levels (low SNR)
13-13.5	drop phase A	Sustained single phase fault
13.5-14	drop phase B (and A)	Sustained two phase fault
14	stop +1 Hz/s ramp at 54.282	
15	dip 20% (all phases) and hold	Sustained marginal voltage level
16-30	ramp frequency down at ~0.66 Hz/s to 45 Hz	Gradual frequency slide
30-31	ramp frequency up at 10Hz/s to 55Hz	Fast +ve ROCOF
31-32.5	45Hz	Sudden frequency step, check locking
32.5-37	ramp frequency down to 0 at -10 Hz/s	Fast –ve ROCOF, low frequencies
37-38	0Hz	
38-39	100Hz	Sudden frequency step, check locking
39-49	100 Hz down to 0Hz at -10Hz/s	Fast –ve ROCOF
49-50	0Hz	
50-60	0Hz to 100Hz at 10Hz/s	Fast +ve ROCOF

Table 2-16 : Waveform 1 to test measurement algorithms

2.11.2   Waveform 2

Time	Description	Purpose / Test
0-1	NOTHING (instrumentation/ADC noise only)	Simulate no connection to network
1-2	44 Hz, 1pu on all phases	Test settling time to a new frequency
2-3	with unbalance added	Adds ripple to 3-phase measurements
3-3.5	with harmonics added	Disturbs all algorithms
3.5-4	with inter-harmonics added	Simulate mains signalling
4-5	with flicker added	Simulate worst case local flicker sources
5-60	ramp frequency up at ~0.2 Hz/s to 55 Hz	Test expected frequency range

Table 2-17 : Waveform 2 to test measurement algorithms

2.12 References for chapter 2

Areva T&D. (2007). Network Protection and Automation Guide. [Online]. [Accessed 26/11/2007]. <http://www.areva-td.com/scripts/solutions/publigen/content/templates/show.asp?P=1228&L=US> Chapter 14

ARW. (2008). Current Transformer Information. [Online]. [Accessed 14/03/2008]. <http://www.arwtransformers.co.uk/CTInfo.htm>

BSI, British Standards. (1998). Electromagnetic compatibility (EMC) – Part 4-15: Testing and measurement techniques – Flickermeter – Functional and design specifications. BS EN 61000-4-15:1998. ISBN 0 580 30061 7

BSI, British Standards. (2000). Voltage characteristics of electricity supplies by public distribution systems. BS EN 50160:2000. ISBN 0 580 35925 5

BSI, British Standards. (2002). Electromagnetic compatibility (EMC) — Part 2-4: Environment — Compatibility levels in industrial plants for low-frequency conducted disturbances. BS EN 61000-2-4:2002. ISBN 0 580 40489 7

BSI, British Standards. (2003). Electromagnetic compatibility (EMC) – Part 4-30: Testing and measurement techniques – Power quality measurement methods. BS EN 61000-4-30. ISBN 0 580 42081 7

BSI, British Standards. (2006). Electromagnetic compatibility (EMC) – Part 3-2: Limits for harmonic current emissions (equipment input current ≤ 16A per phase). BS EN 61000-3-2:2006. ISBN 0 580 48457 2

CDA, Copper Development Association. (2007). Power quality and utilisation guide. [Online]. [Accessed 22/01/2007]. <www.cda.org.uk>

ENA. (1991). ER G59/1. Recommendations for the connection of embedded generating plant to the public electricity suppliers' distribution system.

ENA. (2005). ER G5/4-1. Planning Levels for harmonic Voltage Distortion and the Connection of Non-Linear Equipment to Transmission Systems and Distribution Networks in the United Kingdom.

Hong, Y. & Wang, F. (1997). Investigation of Impacts of Different Three-phase Transformer Connections and Load Models on Unbalance in Power Systems by Optimization. IEEE transactions on power systems, vol. 12 no. 2, pp 689-697.

IEEE. (1991). IEEE Standard Test Procedure for Polyphase Induction Motors and Generators, IEEE Standard 112

IEEE. (2003). IEEE 1547, IEEE Standard for interconnecting distributed resources with electric power systems.

ITI. (2007). ITI (CBEMA) curve application note. [Online]. [Accessed 28/11/2007].

<http://www.itic.org/archives/iticurv.pdf>

Kundur, P. (1994). Power system stability and control. McGraw-Hill. ISBN 0 07 035958 X. Chapter 9.

Laughton, M. & Warne, D. (2003). Electrical engineers reference book. Sixteenth edition. Newnes. ISBN 0 7506 46373. p 34/16.

Moore, P., Carranza, R. & Johns, A. (1996a). Model system tests on a new numeric method of power system frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 2, pp 696-701.

Moore, P., Allmeling, J. & Johns, A. (1996b). Frequency relaying based on instantaneous frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 4, pp 1737-1747.

Mullane, A., Bryans, G. & O'Malley, M. (2005). Kinetic energy and frequency response comparisons for renewable generation systems. Conference on future power systems. Paper O08-01.

Wood, A. & Wollenberg, B. (1996). Power generation, operation and control. 2nd edition. Wiley. ISBN 0 471 58699 4. Chapter 9.

3 Measurement of voltage or current amplitudes & phases

The details of the measurement of voltage or current amplitude/phase within 3-phase power systems are often overlooked. However, accurate measurements are a pre-requisite for any control or protection system. Making these measurements within digital systems requires stages of hardware filtering, sampling and software algorithms. The concepts, ideas, and algorithms employed for these basic algorithms have much in common to those for more complex techniques such as frequency measurement. For this reason, it is useful to examine the "simple" measurement of amplitude and phase of single-phase waveforms in detail, before trying to create an optimal frequency measurement. The shape of the waveforms and design of the measurement stages/algorithm all affect the uncertainty of the answers. In this chapter, algorithms are developed and then the resulting measurement errors are analysed under different conditions of influence qualities (poor power quality, measurement noise, sampling quantisation, low sample rates etc.).

As will be shown in this chapter, there are many different ways to optimise and embellish an algorithm for a seemingly simple measurement of amplitude. Several techniques can be applied in series or parallel, and the interaction of these techniques can be complex. The work in this chapter analyses several techniques and their interactions. An extremely useful product of this work is a suite of algorithms together with a selection matrix which explains the relative benefits and drawbacks of each algorithm versus the computational expense.

The single most useful algorithmic block which re-appears throughout this work is the averaging (or integration) of a quantity over an exact time period. For power system analysis, an obviously useful time period to average/integrate over is a multiple of cycles. This technique was introduced in the MATLAB SimPowerSystems blockset, which in turn is used by Jovcic (2003). This thesis improves upon the MATLAB algorithms for such averaging, both in terms of numerical accuracy and also computational speed, but more importantly introduces the new concept of cascading these averaging stages. It is shown that cascading extra averaging stages of multiples of half-cycles, after an initial single-cycle stage, can be used to almost completely eliminate the interpolation/integration errors due to low sample rate. In addition, the noise reduction properties of such cascaded stages are shown to be superior to other filter types of comparable latency. The cascaded exact-time averaging technique can also be used to build further useful signal processing stages which are developed during the course of this thesis: for example a novel DC blocking technique with zero latency (see section 3.4.4.2),

and an adaptive ripple removal filter (see section 4.3).

During the entire of this chapter, it is important to bear in mind that the amplitude/phase measurements themselves all rely on an estimate of the waveform frequency. In this chapter, the correct frequency is passed directly into the algorithms for most of the simulations. This is possible in a simulation environment where the waveforms are directly synthesised so the frequency is defined and known exactly. In sections 4.1 and 4.6, closed-loop simulations are carried out to determine the latency effects which result due to unavoidable lags in frequency measurement which occur, for example, when a "dead" part of a network is suddenly connected to a "live" network of initially unknown frequency and voltage magnitude/phase.

The path followed in this chapter is:-

- Examination of documented methods.

- Introduce the exact-time-period averaging algorithms in detail, including the substantial improvements made during this thesis, with comparisons to low-pass, Kalman slew-rate filtering techniques.

- Design/analyse the analogue front end and digital pre-processing, including a new DC blocking technique using the exact averaging block.

- Start with the SimPowerSystems "Discrete Fourier" block.

- Modify this for variable frequency operation and computational speed improvement, and incorporate the improved exact-time-period averaging algorithm. Examine the interpolation errors using pure sinusoid waveforms. Extend the averaging algorithm to use second-order interpolation and integration. This is shown to at least half the interpolation error in the absence of noise and harmonic contamination.

- Exploration of second harmonic cancellation techniques.

- Add post-Fourier averaging periods of exact multiples of half-cycles or cycles. This forms a cascade of filters which reject ripple at the fundamental or its harmonics. This novel application is shown to be more effective than second harmonic cancellation, for outputs which can afford a measurement time of 1½ cycles (or more) within the digital environment.

- Explore the use of half-cycle initial Fourier correlations instead of full-cycle correlations, to allow lower quality responses with a ½ cycle digital measurement time.

- Examine the relative dynamic and steady-state performances of the combinations

of techniques against:-

- o Unaliased Harmonics

- o Noise and ADC quantisation

- o Frequency measurement error

- Creation a selection matrix which explains the relative benefits and drawbacks of each algorithm combination versus the approximate computational expense.

- Analysis of the error magnitudes produced due to an inaccurate frequency estimate.

3.1   Documented methods relevant to chapters 3 & 4

There are relatively few publicly available written references on (specifically) the measurement of amplitude and phase in 3-phase AC power systems. Generally, algorithms used in industry are regarded as intellectual property and are not shared in the public domain. There are, however, a growing number of works which encompass the combined problems of frequency measurement and amplitude/phase measurement. Several such references used in chapters 3 & 4 are also relevant to chapter 5 (frequency measurement).

The specific problems addressed within this thesis, particularly low sample rates, very high harmonic content, instrumentation noise and ADC characteristics, do not appear to be addressed together by any known author, although some authors consider one or two of these problems in isolation.

In Johns (1995) the "classic" two-sample and three-sample techniques are described. These methods clearly break down when harmonic contamination such as described in section 2.7 is present. Also described in Johns (1995) are some methods for least-squares fitting of a fundamental component, some harmonic components and a decaying DC term to AC waveforms. In the final analysis, this boils down to a set of Fourier transformations plus a further analysis of a decaying DC term. Also in Johns (1995) are interesting differential-equation based algorithms where the differential equations are loaded with measured parameters of transmission lines. This is really an impedance measurement technique rather than an amplitude/phase measurement. The three-phase sampled measurements are processed to determine the RLC impedances, which are particularly relevant during faults. These techniques are primarily aimed at detecting faults in high-voltage transmission lines, potentially with sub-cycle measurement times. This is relevant in a transmission system where the protection is fast "unit" protection, the waveforms are generally clean sinusoids, and fault currents cannot be tolerated for many cycles before transmission lines sag or melt. Thus these techniques are interesting but not

particularly relevant to the problem addressed by this thesis, which is concerned with LV systems, graded protection systems with longer tripping times, and dirty waveforms of voltage and current.

Petrovic (2000) described using extremely slow (4 to 96Hz) but extremely high precision ADCs to make amplitude measurements. The assumption is that the power system is "inert" and therefore the measurement is made over many cycles. This is not the scenario presented in this thesis so the reference is not pursued further.

Aghazadeh (2005) presents a method for amplitude measurement based upon Kalman filters. (Also in this paper is a proposed zero-crossing based frequency measurement referred to in chapter 5). The method shows "unavoidable" transient deviation errors of up to 0.5pu when frequency changes, which is unacceptable.

Lin (2005) describes an interesting algorithm (for both amplitude and frequency measurement) based upon adaptations of wavelet transforms. The sample rate used is also suitable, at 600Sa/s, and this method might be worth more analysis in the future. No noise or ADC quantisation is applied, however, and the scales on the graphs do not allow a good analysis of the performance. His proposed method is compared to a Fourier analysis (which is the proposed base measurement used in this thesis), but this appears to have been coded very poorly (either accidentally or deliberately) to give bad results for comparison.

Some of the most relevant references found to date are actually the standard library blocks within the MATLAB SimPowerSystems blockset. These are described further in section 3.5 and form a useful starting point for the work of this thesis. Also, Jovcic (2003) presents a PLL design which includes a second harmonic cancellation scheme which has benefits in certain scenarios. This is discussed further in section 3.8.

Apart from the original SimPowerSystems blocks and the work taken from Jovcic, all the methods presented in chapters 3 & 4 are necessarily novel work carried out during the course of this thesis, due to the scarcity of relevant references found.

3.2   Averaging signals over exact time periods

This section explains how to average a signal over an exact time period within a digital system with a fixed frame rate. This technique proves to be useful in many ways, for example:-

- During Fourier correlations or RMS measurements of AC waveforms, the mathematical equation for the measurement requires definite integration (i.e. averaging) over exact periods which are multiples of cycles (or half cycles).

- Ripple can be present on measured signals. This ripple is often at the fundamental frequency f or its second (or higher) harmonic. Desired averaging times are often therefore, multiples of $1/f$ seconds, or multiples of $1/(2f)$ seconds if the lowest expected ripple frequency is at twice the fundamental. Later, in section 4.3, a novel ripple-removal filter is introduce which can use averaging times not necessarily equal to integer multiples of $1/(2f)$ to remove ripple at inter-harmonic or sub-harmonic frequencies.

The problem with exact-time averaging within a fixed-frame-rate system is that the desired time period to average over is normally not a convenient integral multiple of the frame time. Thus, interpolation techniques are required to address the "part-sample" problem. This is the reason that some specialised digital relays and power quality meters actually adjust their system clocks in such a way that an exact number of samples occur during one cycle, as described in section 2.9.

Within fixed-frame-rate systems, this technique is not available and an interpolation technique must instead be used. SimPowerSystems already has such a block which addresses this problem (with limitations), called "Discrete Variable Frequency Mean value". The idea of this block is shown below:-
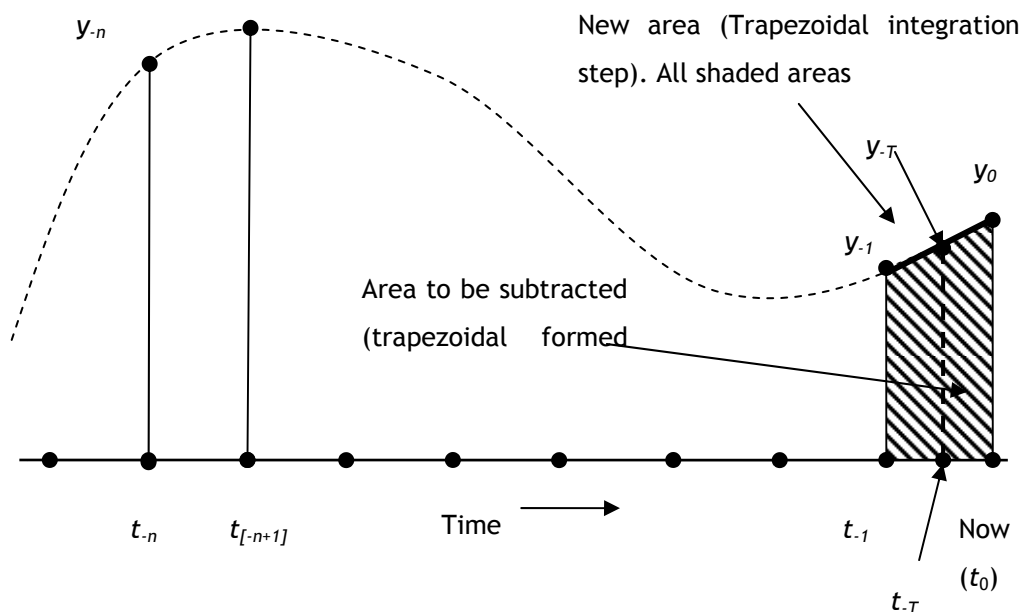
Figure 3-1 : SimPowerSystems "Discrete Variable Frequency Mean value"

74

The block works as follows:-

- Each new sample comes in as y_0 at time t_0.

- A continuous trapezoidal integration process accumulates the integral formed by y_{-1}, y_0, t_0, and t_{-1}.

- The value of the integration is stored in a rolling buffer, big enough to store enough samples to cope with the longest required averaging time.

- The old value of the rolling integration at time t_{-n} is pulled from the buffer, and subtracted from the accumulated integration at t_0. This reveals the definite integral from t_{-n} to t_0.

- This is then corrected to give the definite integral from t_{-n} to t_{-T}, where T is determined by the exact non-integer number of frame times which fit into the desired averaging time. The correction is applied by linearly interpolating between y_{-1} and y_0, and using this to calculate the value of y_{-T}, and thus the area of the trapezoid formed by y_{-T}, y_0, t_0, and t_{-T} can be calculated and subtracted from the integral from t_{-n} to t_0.

- Thus, the final output is the definite integral from t_{-n} to t_{-T}, which can be converted to an average by dividing by the desired averaging time (t_{-T} - t_{-n}).

This algorithm, as implemented within SimPowerSystems, has a few shortfalls which can be significantly improved upon. These are described below:-

1. The output is "out of date" by between 0 and a full sample. The value depends upon the fraction of a full-cycle (T in the diagram above), which must be back-tracked to. This could be a disastrous effect if the averaging block is used inside a PLL which is being used, for example, as part of an inverter drive. It could introduce up to a full frame lag into the control system, severely affecting performance. Worse still, the lag is variable depending upon the resulting value of T, which means that any control system dynamics would be varying "at random" as T varied. The solution is to move the interpolation to the samples which are the oldest, so that the newest sample is used without interpolation. The measurement is thus always exactly coherent with the most recent sample, and thus also the processor clock.

2. To reduce interpolation errors, the linear interpolation and trapezial integration can be replaced using a 2nd-order polynomial fitting technique.

3. If a signal with any DC component is input, over time the integrator will eventually

lose precision and saturate. This is generally not a problem in the simulation environment where relatively few seconds elapse and the arithmetic is often 64-bit precision. In a real-time environment, run-times may be tens of thousands of hours, using 32-bit precision arithmetic (or less). Additionally, to average a result of phase, the risk of integrator wind-up is serious since the input phase must be un-wrapped before being averaged. To enable the algorithms to work in these cases a system of twin integrators is required with a "tick-tock" type "reset and use" system is required. This technique is described in detail below.

3.2.1 Improving the latency of the exact-time averaging

To address the first point, the algorithm can be modified in the following way:-
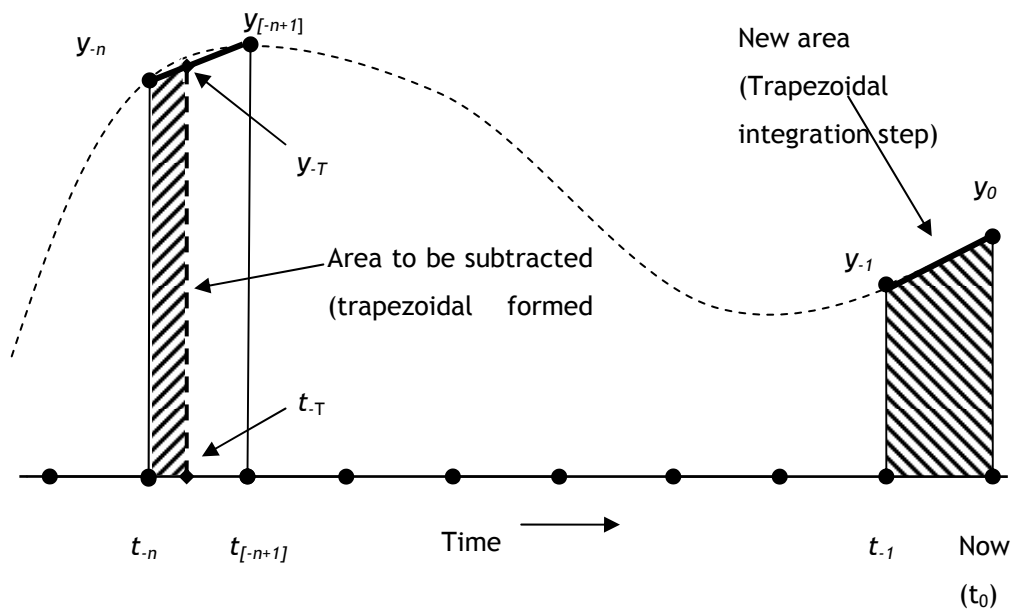
Figure 3-2 : Improved exact-time averaging technique

The interpolation and subtracted part-sample area can be moved from the most recent sample to the oldest, as shown in Figure 3-2. This means that the result is always coherent with the frame clock. The other calculations are basically identical. A slight complication is that now two buffers are required. The buffers are the same length. One buffer contains the rolling integral values as before. The second buffer contains the sampled values y_0 to y_{-n}, and has a "two-tap" output so that the values y_{-n} and $y_{[-n+1]}$ can both be retrieved for the calculation of the area to be subtracted via interpolation to y_{-T} The "two-tap" buffering algorithm can be implemented in Simulink by the following piece of code:-

Variable discrete delay with two taps

Andrew Roscoe, July 2005
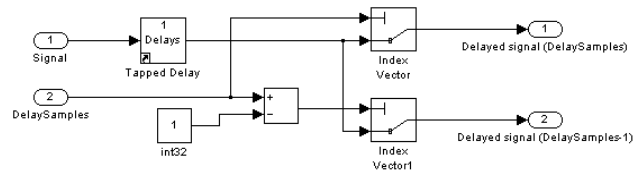
Figure 3-3 : Buffer (2 taps) for linear interpolation

However, when compiled into C code for execution on target hardware platforms (e.g. the Infineon TC1796 microcontroller) using the MATLAB Real-Time-Workshop module, this block results in a "MEMCOPY" assembler instruction which can be very costly in terms of CPU time, with execution time increasing with increasing buffer size. To address this, MATLAB SimPowerSystems contains an S-Function version of a "single-tap" buffer block called "Discrete Variable Transport Delay". This executes orders of magnitude faster, and the execution time is independent of the buffer size. During this thesis, new delay/buffer S-Functions were produced. These use a fixed memory space and moving pointers both to add and to tap off the input and output data. Optimised codings for the simple "one-tap" buffer, the "two-tap" buffer and also the "three-taps" buffer (required for 2nd order interpolation presented below) have all been generated. The delay buffer S-functions produced during this thesis execute up to 3 times faster than the equivalent MATLAB SimPowerSystems S-function buffer block "Discrete Variable Transport Delay". For details on the benchmarking of these S-Functions and example S-Function code ("c" and "tlc" files), the reader is directed to Appendix G.

The overall algorithms (Simulink code plus S function calls) for the 2nd order averaging blocks are presented in the next section. The 1st-order algorithms are very similar but simpler.

3.2.2 Extension to 2nd order interpolation

The entire algorithm can be extended to use a 2nd order quadratic technique to minimise the interpolation errors. The idea is simply to use 3 points to fit a quadratic and thus to obtain a more accurate measure of the area/average.
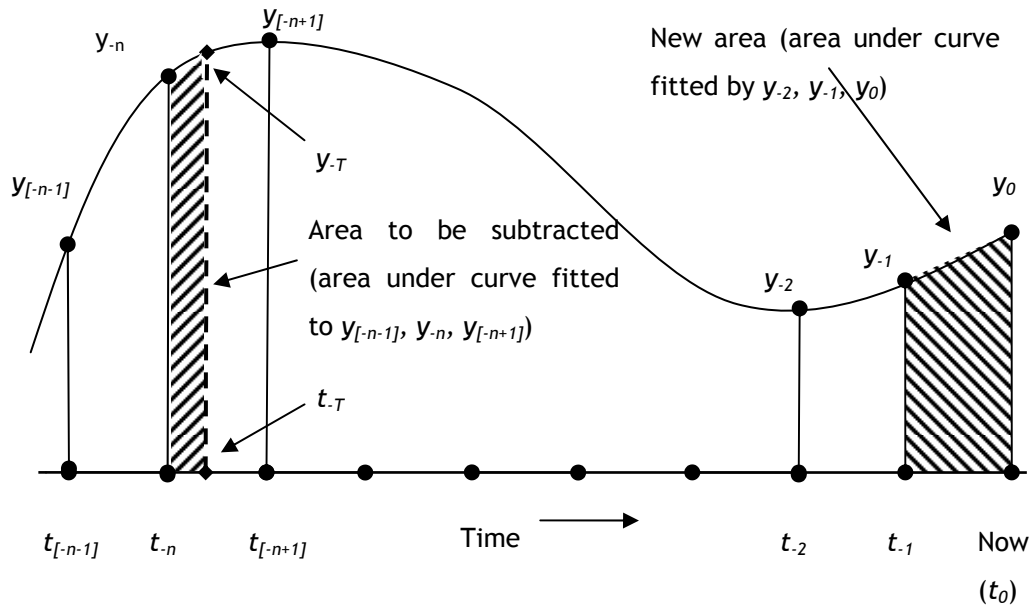
Figure 3-4 : 2nd order exact-time averaging technique

Figure 3-4 shows how this is done. The new area to be added to the integration buffer at each sample is computed by first fitting a quadratic polynomial to the most recent three samples y_{-2}, y_{-1}, and y_0. The area from t_{-1} to t_0 can then be calculated by evaluation of the integration of the fitted curve. Similarly, at the "old" end, the area to be removed is calculated by fitting a quadratic to the points $y_{[-n-1]}$, y_{-n}, and $y_{[-n+1]}$. Note, this requires buffering the *y* values for one sample longer than was done in the linear interpolation case. This is required so that the curve fitting at the old end to remove area from a particular segment uses the same set of three points as were used to add the area of the original segment. Experiments show that using the points y_{-n}, $y_{[-n+1]}$ and $y_{[-n+2]}$ to calculate the area to be subtracted does not work as well.

This quadratic fitting initially sounds computationally expensive, but it can be achieved without using any square root functions. The overhead over and above the linear interpolation case is that one buffer must be 1 sample longer, and that there are a few more multiplication and addition/subtraction operations. The calculation details are shown in the next section.

3.2.3   Avoidance of integrator wind-up, and algorithm detail overview

The SimPowerSystems block "Discrete Variable Frequency Mean value" algorithm includes a rolling integrator which runs forever. To avoid this winding up and/or losing precision, an

arrangement of twin integrators in a tick-tock arrangement has been implemented. To explain this, it is easiest to present the actual Simulink coding. The 2nd order block is used as an example.
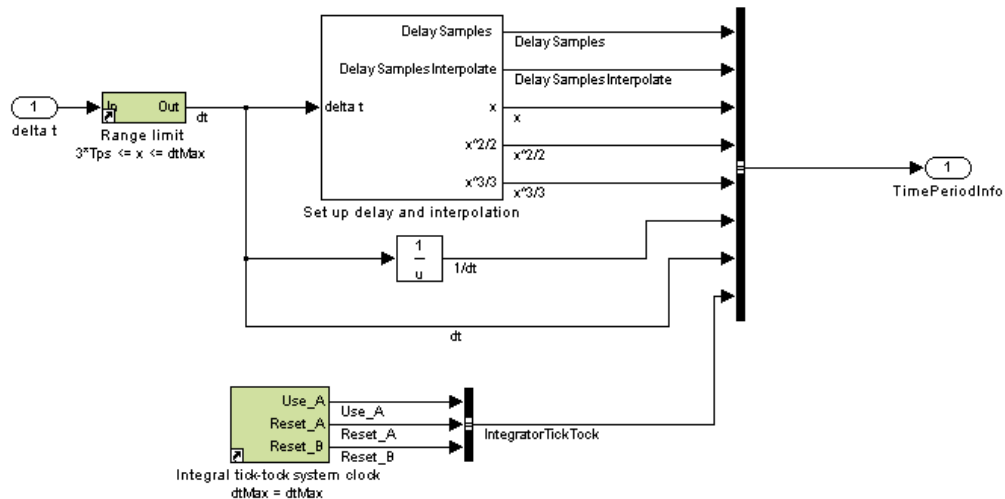
Figure 3-5 : Re-usable pre-calculations for 2nd order averaging

Figure 3-5 shows the re-usable pre-calculation signals for the buffering, curve fitting and interpolations, plus timing signals for the tick-tock integrator. The input to the algorithm is the time over which the averaging is required. This block is re-usable in that the *TimePeriodInfo* output can be passed to many blocks which need to average different signals but over the same time period. The detail inside the "Set up delay" block is shown below in Figure 3-6.

The outputs from this block are:-

- the number of integer samples to delay inside the integration buffer, *DelaySample*)

- the number of integer samples to delay inside the sample (*y*) buffer, *DelaySamplesInterpolate*

- The precalculated values of *x*, $x^2/2$, and $x^3/3$

Calculate the value x required for 2nd order interpolation

x is the time after the middle sample (y2) of the three used for curve fitting.
The sample y2 is assigned time=0
The sample y1 is assigned time=-1
The sample y3 is assigned time=1

Andrew Roscoe, 2007

NOTE. The DelaySamples for the interpolation goes back one further.
This is so that the interpolation for the extra area removal works with the same
three samples that the area addition inside the original integration worked with.

Therefore
x = DelaySamples - Idealsamples
where Idealsamples is the number of non-integer samples in the past which we require to interpolate to
and DelaySamples+1 is the actual (integer) number of samples which are held in the buffer.
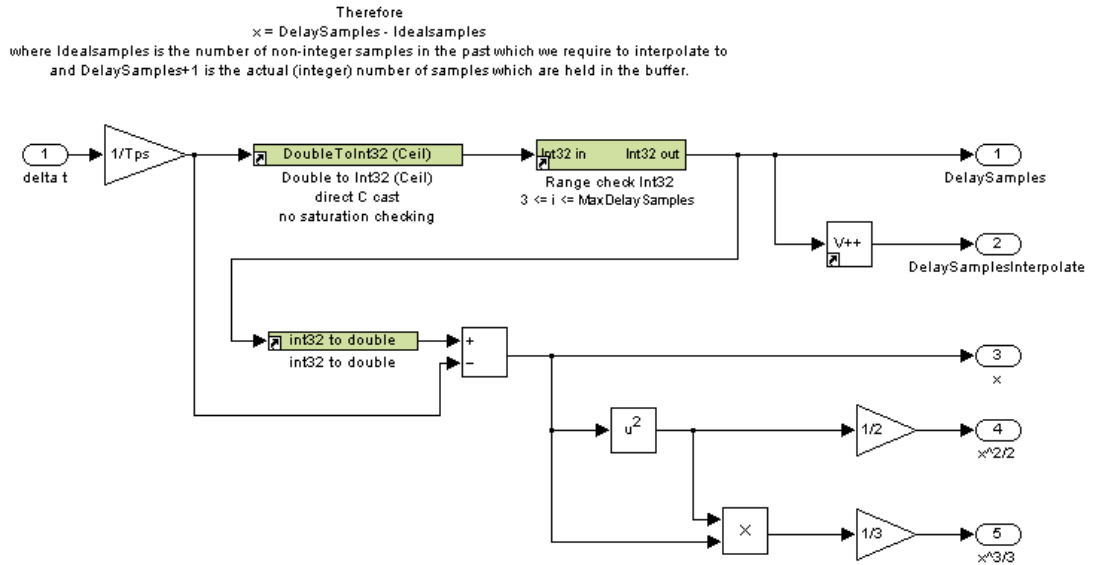
Figure 3-6 : Re-usable pre-calculations for 2nd order averaging, detail

Note that to simplify the arithmetic, the quadratic curve fitting and interpolation is normalised to operate as if the three points were at [-1, *y1*], [0, *y2*] and [1, *y3*]; i.e. with a frame time of 1 second. Thus *x*, as output from this block, is the time value at which the interpolation is required, and is in the range 0<*x*<1. The final answer is corrected (de-normalised) by multiplying by the actual frame time at the very end.

Clock for Tick-tock definite integral implementation

Andrew Roscoe, 2007
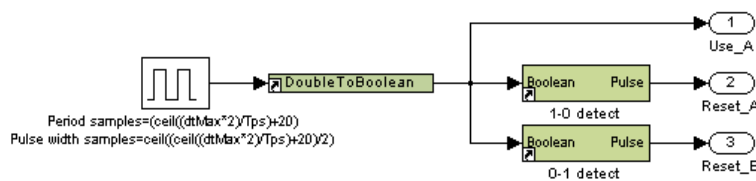
Figure 3-7 : Re-usable pre-calculations for 2nd order averaging, tick-tock timing

Figure 3-7 shows the final part of the pre-calculation. The primary output is a Boolean output which signifies whether to use path A (or path B) of the twin integrators (see Figure 3-8). Also, there are two reset signals which reset the integrators when they become

inactive. The signal timing is such that before each integrator output is used, it must be fully loaded and settled with valid signal values after being reset. The fastest tick-tock clock can therefore be *dtMax**2 where *dtMax* is the longest allowed averaging time (and hence also sets the required buffer size). In practice, a slightly slower clock is used to guarantee clean handovers between the pair of integrators.

The remainder of the 2nd order averaging algorithm is now described. The inputs to the block are the signal to be averaged, and the *TimePeriodInfo* data from Figure 3-5.

Figure 3-8 : 2nd order exact-time averaging block, tick-tock system
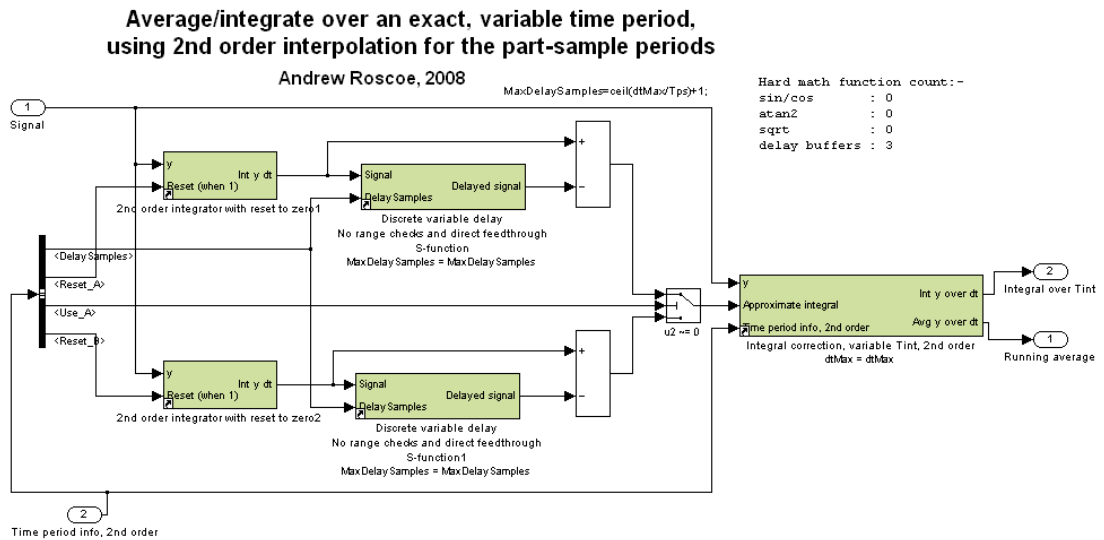
The integration has two paths, A and B. Each is used for a certain time, then reset, then preloaded with valid signal data, then used, etc. The two paths together produce a seamless output with no risk of integrator wind-up.

Figure 3-9 : 2nd order exact-time averaging block, 2nd order integrator

81
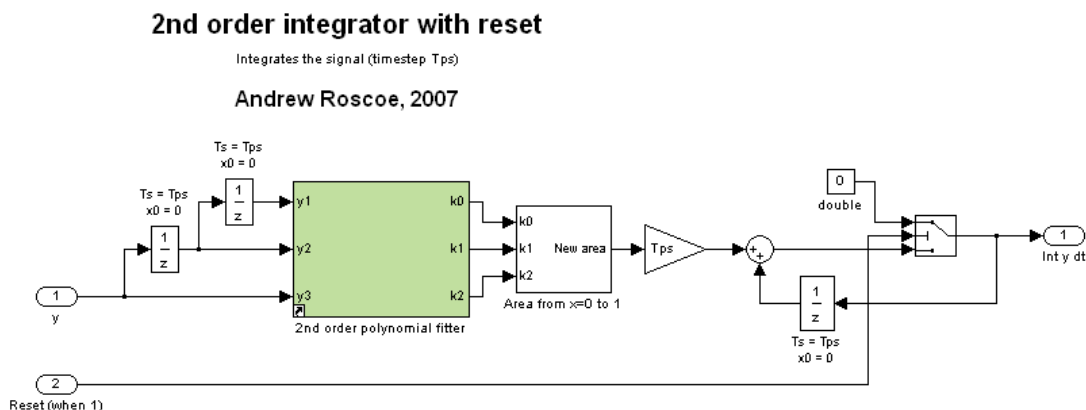
The integrator accumulates the integral of the signal by curve fitting and evaluation of the integral of this curve. For computational simplicity, the three most recent sampled points are here considered to lie at $[-1, y_{-1}]$, $[0, y_0]$ and $[1, y_1]$. The correction for this normalisation is carried out later in Figure 3-9 by the multiplication by Tps (the frame time).
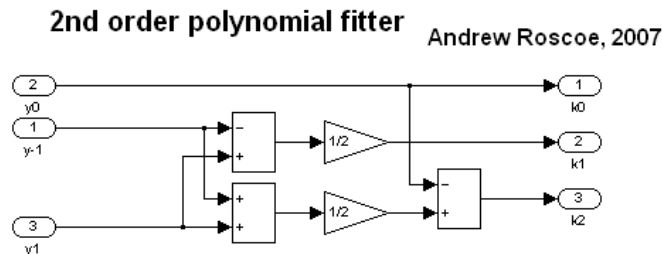
Figure 3-10 : 2nd order exact-time averaging block, quadratic curve fitting

The quadratic curve fitting requires only simple arithmetic, as shown in Figure 3-10, which evaluates:

$$k0 = y_0$$

$$k1 = \frac{y_1 - y_{-1}}{2}$$

$$k2 = \frac{y_1 + y_{-1}}{2} - y_0$$

Where *k0*, *k1* & *k2* allow *y(t)* to be estimated within the region -1<*t*<1, by using

$$y = k0 + k1 \cdot t + k2 \cdot t^2$$

(3.1)
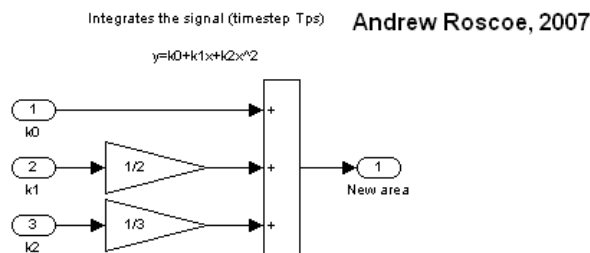
Figure 3-11 : 2nd order exact-time averaging block, quadratic curve integration

The new area to be integrated, between *t*=0 and *t*=1 (see Figure 3-4) can be calculated from *k0, k1, k2* with only 2 multiplications and 2 additions, as shown in Figure 3-11. This numerically evaluates the integral of equation (3.1).

82

$$\int_0^1 \left(k0 + k1 \cdot t + k2 \cdot t^2\right) \cdot dt = \left[k0 \cdot t + \frac{k1 \cdot t^2}{2} + \frac{k2 \cdot t^3}{3} \right]_0^1 = k0 + \frac{k1}{2} + \frac{k2}{3}$$

The output of the blocks described in Figure 3-9 to Figure 3-11 is a running integration. There are two sets of these blocks forming the tick-tock system, plus a buffer in each path which allows the definite integral to be calculated (see Figure 3-8). The final piece of the algorithm is the correction for part-sample effects, by the "Integral correction" block, which requires a final (3rd) delay buffer.
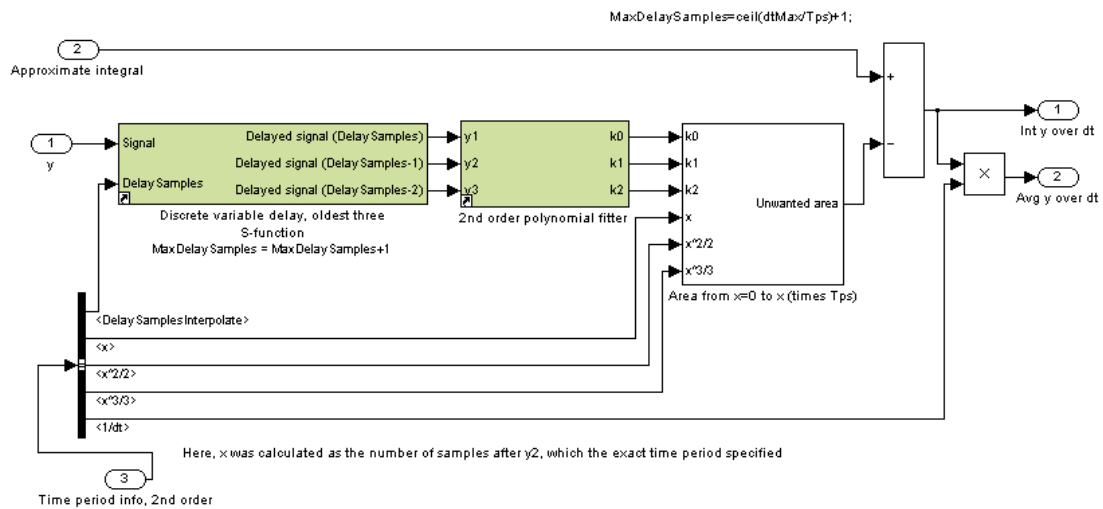
Figure 3-12 : 2nd order exact-time averaging block, definite integral calculation

Figure 3-12 shows the code used to carry out the process described in Figure 3-4. The final piece to be described is the detailed calculated of the area to be removed for correction of the part-sample period not required (Figure 3-13). It can be seen that this integration only requires simple calculations, by re-using the pre-calculated values of x, $x^2/2$, and $x^3/3$.
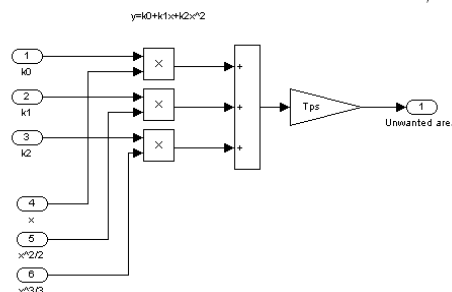
Figure 3-13 : 2nd order exact-time averaging block, definite integral correction area

The final results are the integral (and average) of the signal value over the exact time period desired. The linear interpolation coding is similar but slightly more straightforward due to the use of linear interpolation instead of quadratic curve fitting. With linear interpolation, the calculations are simpler, and also more of them can be pre-calculated and shared amongst many blocks using the same averaging period.

3.2.4   Averaging of phase

Averaging of variables which contain phase information requires special care. The input phase to the averaging algorithm is generally (but not necessarily) in the range $-\pi<\varphi<\pi$. Great case must be taken so that the phase wrapping effect at the $\pm\pi$ boundary does not corrupt the averaged result. The averaged result should also always be in the range of $-\pi<\varphi<\pi$. Extending the algorithm from that of Figure 3-8 to cope with phase, results in the algorithm of Figure 3-14 and Figure 3-15. The input phase must be "un-wrapped" before integrating, and re-wrapped before being output. Loss-of-precision errors are avoided by the tick-tock system which not only resets the integrators but also resets the un-wrapping code which otherwise can easily saturate when executed in real-time for many seconds/hours.
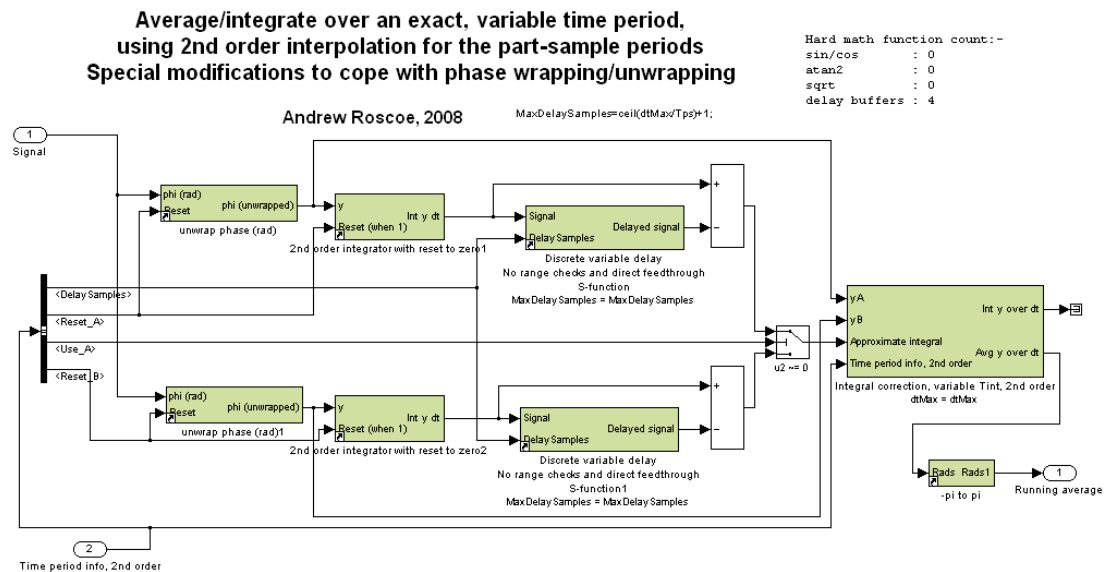
Figure 3-14 : 2nd-order exact-time averaging extended to variables measuring phase (a)
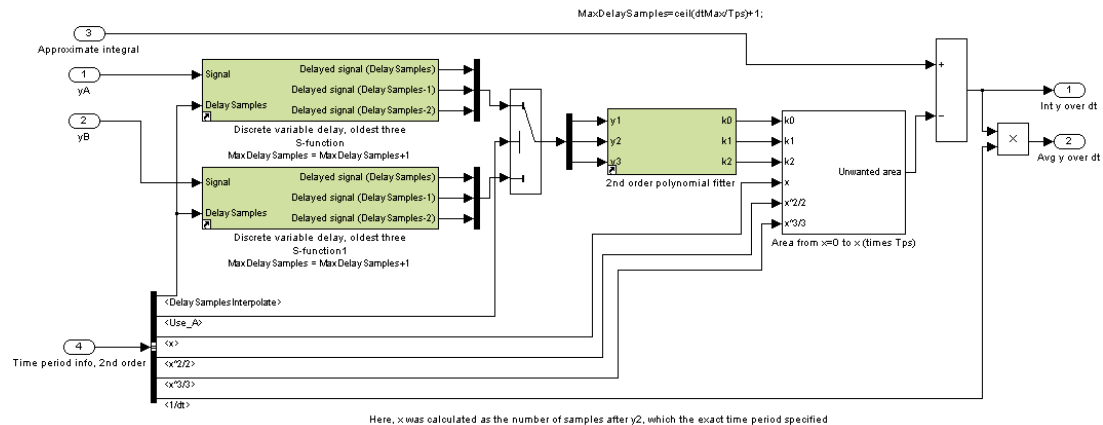
Figure 3-15 : 2nd-order exact-time averaging extended to variables measuring phase (b)

3.3 Comparison of exact time period averaging techniques with low-pass/Kalman filters and slew rate filters

Ever-present problems in measurements of AC power system parameters are:-

- Noise

- Ripples due to sampling, harmonic, and aliasing effects

To cope with these effects, several filtering techniques are considered during this thesis

- Low-pass filtering

- Kalman filtering

- Slew-rate limiting

- Exact-time averaging

Some references including Dash (2000) advocate the use of Kalman filtering. This was briefly investigated. At the core of the Kalman filter, however, is an algorithm which is a tunable low-pass filter (Welch, 2001), which can be adapted to different noise conditions. Low-pass filters have been investigated thoroughly, and this thesis proposes that the use of exact-time averaging techniques is more suitable within the power systems domain than the low-pass filter. Low-pass filters and Kalman filters can be simpler to implement in real-time, being IIR filters which require only a single state to be stored in memory. The averaging techniques described in section 3.2 are FIR filters which require buffering of data streams in memory. It is proposed that these buffers are nowadays relatively easy to implement on even the smallest modern microcontrollers, where many kB of RAM is available with acceptable access speeds. Traditionally, this has been a limitation on historic computer systems with limited RAM memory.

To illustrate the relative performance of low-pass, exact-time-averaging, and slew-rate filters, a simple Simulink model has been created. This allows the following four scenarios to be presented to filters:-

- A step function from 0 to 1 at t=1 second

- An impulse of value 1 at time t=1 second

- Gaussian noise, RMS value 1

- A ripple (sine wave) at a worst frequency for the exact-time averaging algorithms, which results in the biggest interpolation errors. These worst frequencies are such that the desired averaging time equates to $n+\frac{1}{2}$ sample periods where n is any integer.

The frame rate is set at 500 Sa/s (nominally 10 samples/cycle for a 50Hz input waveform). A worst input frequency for ripple in this case is one where only 9.5 samples occur during a cycle, i.e. 52.632Hz. A worst time period for averaging is therefore a multiple of 1/52.632 second.

The first analysis is to compare the 1st-order exact-time period averaging block with the original SimPowerSystems "Discrete Variable Frequency Mean value" block. In the case shown below, the step function is presented to the two blocks, which both have averaging times set to 1/52.632 seconds (9.5 samples). It can be seen in Figure 3-16 that the SimPowerSystems block outputs have additional ½ cycle latency relative to the new 1st order averaging block.
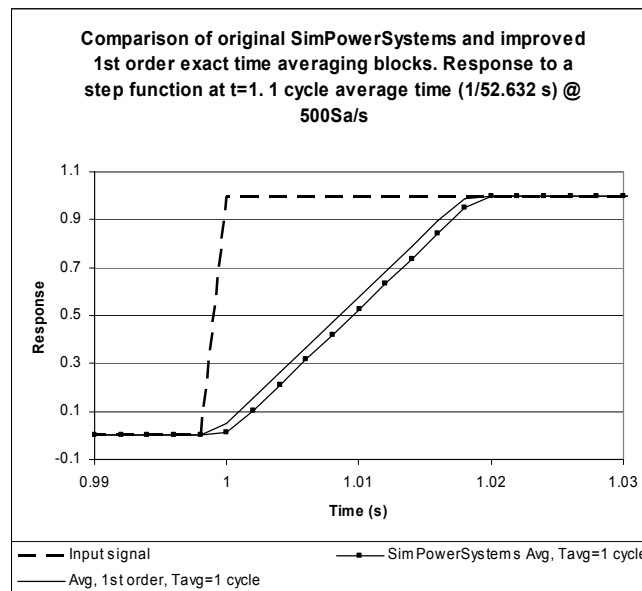
Figure 3-16 : Improved latency of the exact-time integration block over the SimPowerSystems block

Next, a wider range of filters are further compared for performance. The desired time to obtain a settled result after a transient (step function), is set at 5 cycles (5/52.632=0.095 seconds):-

- Low-pass filter, with a "5RC" time of 0.095 seconds, hence F_c=8.38Hz by {B.5}

- Exact-time period average block, 1st order, set to 0.095s

- Exact-time period average block, 2nd order, set to 0.095s

- Slew-rate limiter, with rate limited to 1/0.095s = 10.526/s

- Cascaded pair of 1st order exact-time period average blocks, set to 2.5 cycles and 2.5 cycles averaging time periods (0.0475s each)

- Cascaded pair of 1st order exact-time period average blocks, set to 1 cycle and 4 cycle averaging time periods

- Cascaded trio of 1st order exact-time period average blocks, set to 1 cycle, 0.5 cycle and 3.5 cycle averaging time periods

3.3.1   Step response of filters

The first comparison is the response to the step function at t=1 second. Analysis of Figure 3-17 and Figure 3-18 shows that all of the exact-time average filters, and the slew-rate filter, are fully settled by the required time (t=1 second + 0.095 seconds), or a sample or two after it. This would be expected for the FIR filters. The low-pass filter, however, despite its faster initial rise-time, is only 99.3% (100*(1-e^{-5})) settled at this time.
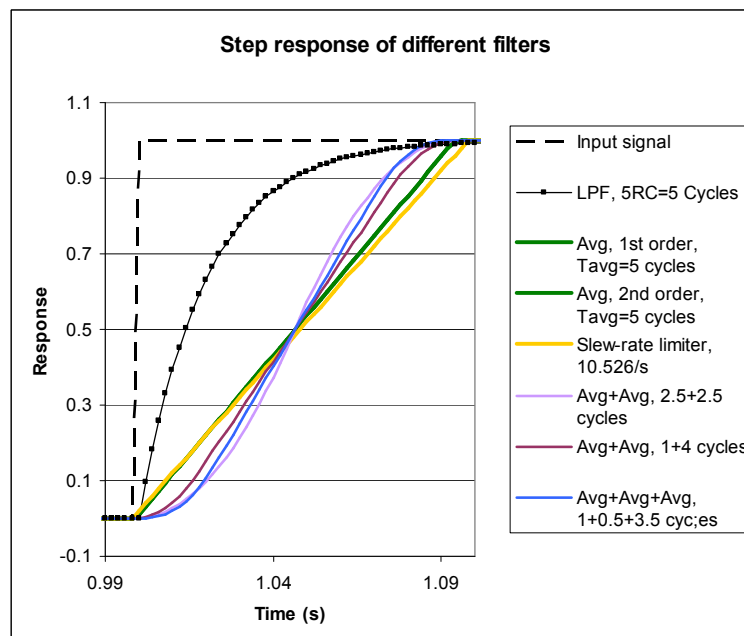
Figure 3-17 : Step response of different filters

87
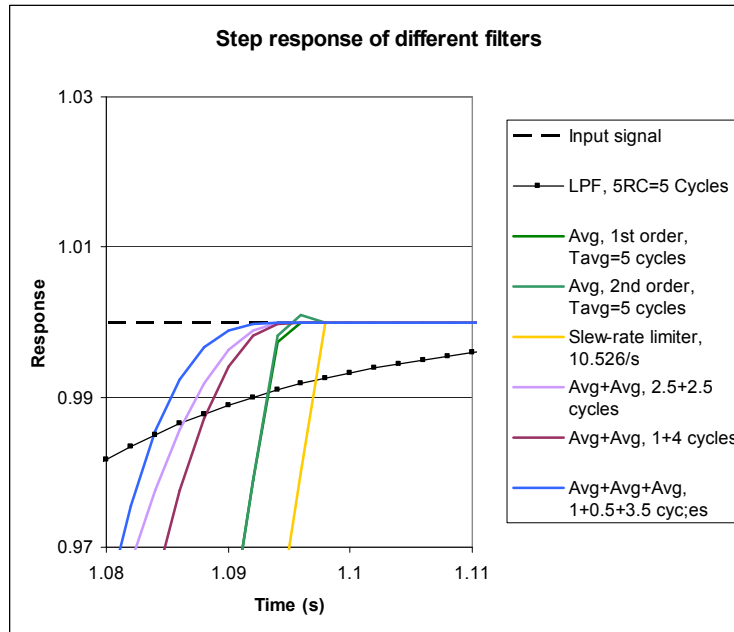
Figure 3-18 : Step response of different filters, zoomed in

3.3.2   Impulse response of filters

The next analysis of the filters is the response to an impulse function at t=1 second, which is made up of a single sample of amplitude $1/T_s$=500, where T_s is the sample time (frame time) of 1/500 s.
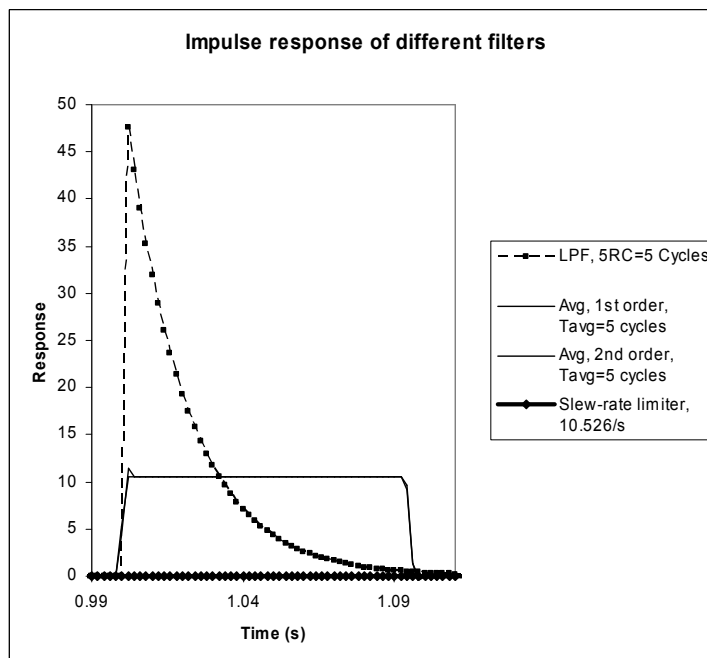
Figure 3-19 : Impulse response of different filters

Figure 3-19 clearly shows that the low-pass filter has the largest peak impulse response. By comparison, the other filters have a much smaller response. This indicates that sampled spikes of noise will be better smoothed by the averaging and slew-rate filters, when the filters are configured for the required settling times due to transients (step functions). The difference between the low-pass filter and the averaging filters is that the response of the averaging filter is smeared over a longer time, with equal weighting to all the samples in this averaging timeframe, whereas the low-pass filter response is weighted towards the more recent samples. Since the low-pass and averaging filters are both linear, the area under the curves for these filters in Figure 3-19 is 1, or very close to 1 due to interpolation errors. The value 1 here is the same as the area of the input impulse waveform of $1/T_s$ for T_s seconds. The slew-rate response is quite different due to its non-linearity via the clipping effect. In the case shown here its output peaks at only $10.526*T_s=0.021052$, due to the slew rate limitation.

Due to the filter weightings (the shape of the impulse response), the low-pass filter gives a higher weighting to more recent inputs than to older inputs. The averaging filter weighting is constant for the defined averaging period. A low-pass filter (or Kalman filter) is thus the correct approach when the desired output is "what is the system doing right now?". The averaging filter, however, is more appropriate to use when the desired output is "what has the system been doing for the last *x* seconds?". The Kalman filter, being an extension of a low-pass filter, is thus ideal for tasks such as spacecraft position estimation, which was the first implementation of such a filter. However, within AC power systems the measurement of an AC waveform has no real concept until at least ½ a cycle can be accumulated. The question is thus "what has the waveform been doing for the last ½ a cycle or N cycles", and therefore the averaging filter is more suitable to this task. This does not mean that the answer is only updated every ½ or N cycles. The answer can be updated many times per cycle, but the answer at any time refers to the shape of the waveform over a previous amount of time, and not to the instantaneous sampled voltage or current.

An interesting observation here is the concept of the Heisenberg uncertainty principle, which refers to the way that it is impossible measure a particle's position or velocity exactly when it behaves as a wave function. An AC voltage or current waveform is also a wave, and as such cannot be measured at any single point in time. It must be analysed over a period of time to form a "most likely estimate" of what that wave is actually doing. Such academic statements are in practice backed up by conclusions from field testing of relays such as in Moore (1996a), which states that, although academic researchers often strive for the fastest-responding measurement, brief sub-cycle transients and events must not lead to spurious tripping.

3.3.2.1   Impulse response of cascaded averaging filters

A further relevant set of results is obtained by comparing the (convolved) impulse response of various cascaded averaging filters.
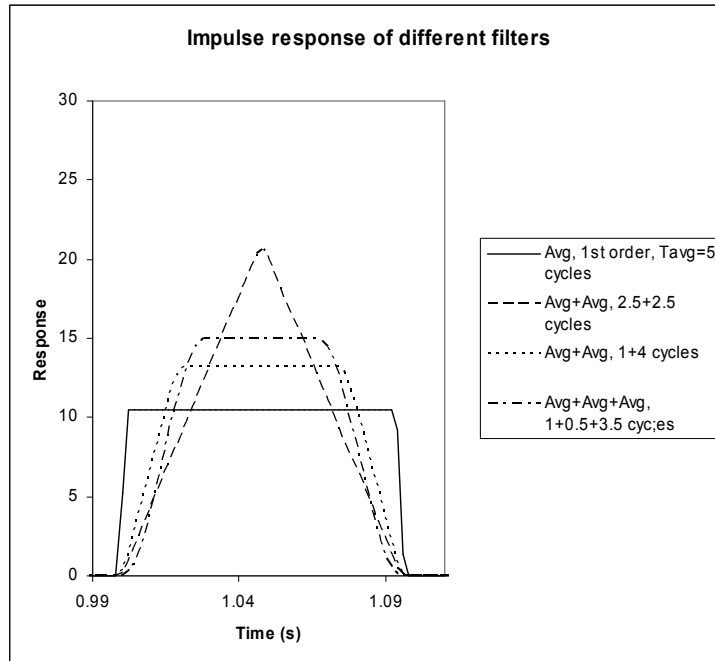
Figure 3-20 : Impulse response of different cascaded averaging filters

All the filter combinations in Figure 3-20 have the same total setting time of 5 cycles, but the cascading is different. This is highly relevant for later sections such as section 3.9 and section 4.3, where a 1-cycle Fourier correlation (involving a 1-cycle averaging) can be followed up by a ½ cycle averaging, and then by subsequent stages of averaging to provide the optimum response to ripple and noise. The peak magnitude of the impulse response for a lone average filter is proportional to $1/N$ where N is the number of samples (or the length of time) over which the average is taken. It can be shown that the peak magnitude of the response of a cascaded pair of averaging filters will be proportional to $1/\max(N1,N2)$ where $N1$ and $N2$ are the lengths of the two filters in samples (see appendix B.4). Thus, for the cascaded pair of filters of length 2.5 cycles each, the peak magnitude is twice that of the lone 5-cycle filter. The other cascaded combinations provide intermediate results.

3.3.3   Frequency response of digital filters

To reinforce the statements of section 3.3.2, the bode plots for the digital low-pass and digital averaging filters can be compared. Below are shown the zero & pole positions and the bode plot for the low-pass filter with "5RC" set to 0.095 seconds (F_c=8.38 Hz by {B.5}).

90
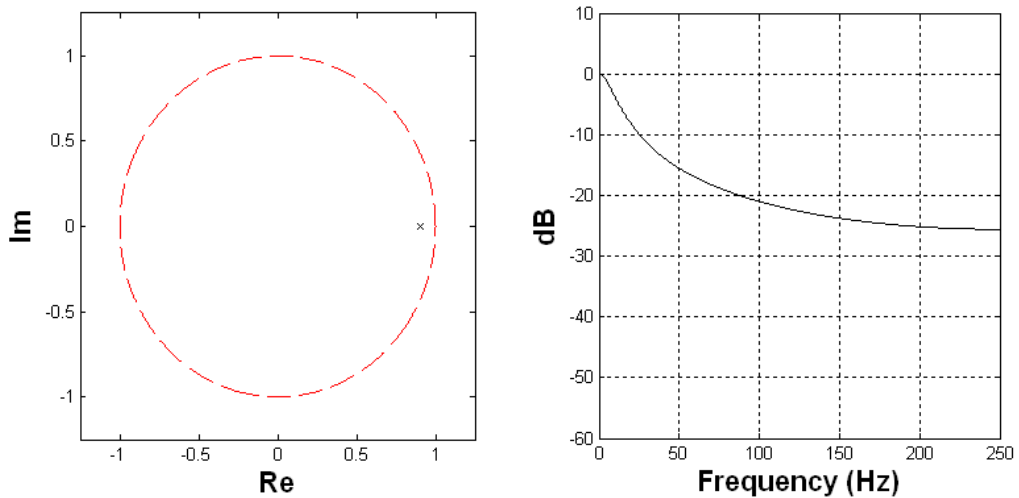
Figure 3-21 : Single pole and bode plot for low-pass filter with Fc=8.38 Hz ("5RC"=0.095s), T_s=1/500 s

The low-pass filter shows relatively poor attenuation of signals above 100Hz, when compared to the equivalent plot (below) for the single-stage averaging filter with 47 samples of averaging (equivalent to averaging over 0.094s).
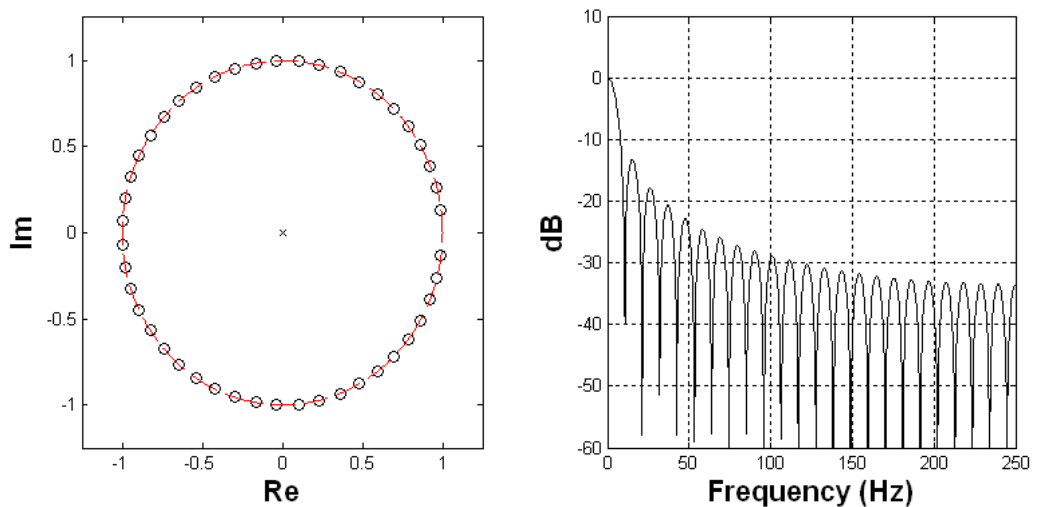
Figure 3-22 : Poles, zeros and bode plot for averaging filter with 47 samples, T_s=1/500

This is a filter with 47 zeros and 47 poles. The poles are all at 0+0j, and the zeros are scattered around the unit circle. As such, it forms a comb filter with many notches.

Figure 3-22, when compared to Figure 3-21, explains why the averaging filter is much more effective at removing noise than the low-pass filter. The averaging filter has >5dB better attenuation across the range, particularly at the notch frequencies.

Referring back to section 3.3.2.1, the case of the two cascaded averaging filters can also be analysed. This is done by using 2 off, 23-sample averaging filters in cascade (0.092s total response time). When transformed into the z domain, this results in 46 poles at 0+0j, and 46 zeros on the unit circle (see Figure 3-23). As distinct from Figure 3-22, however, the zeros form only 23 distinct zeros, in 2 sets of identical pairs overlaying each other. Thus, the number of notches is halved, but the attenuation between them is improved. This filter combination shows up 30dB better attenuation than the low-pass filter for frequencies approaching half the Nyquist frequency. Thus, a cascaded pair of equal-length averaging filters is much, much better at attenuating unwanted Gaussian (broadband) noise signals than a low-pass filter, when the averaging filter latency (response) time is set equal to the low-pass filter "5RC" time.
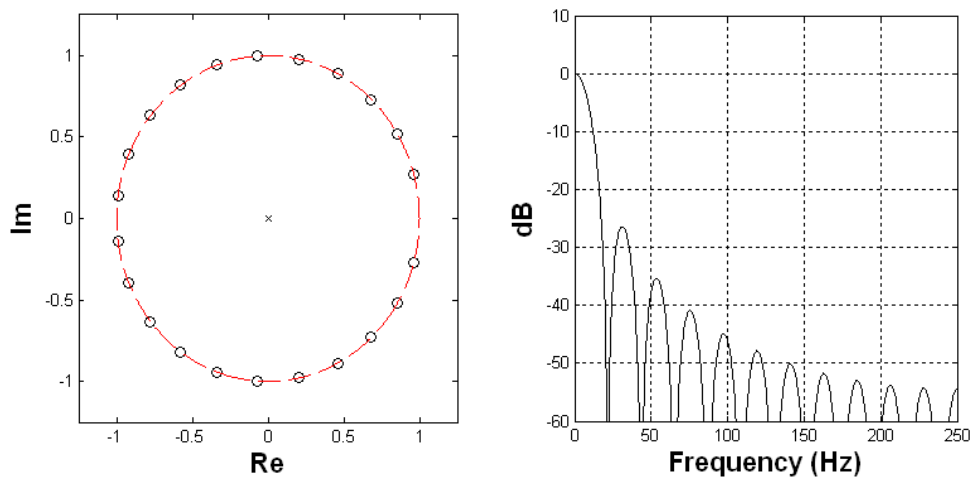
Figure 3-23 : Poles, zeros and bode plot for 2 cascaded averaging filters with 23 samples, Ts=1/500 s

3.3.4   Noise rejection of filters

In the case of a single averaging filter, length 5 cycles, an individual noise spike entering the filter immediately affects the output by a weighting which then stays constant for the duration of the filter length. After this time, the noise spike effect is then totally removed from the filter output. The bandwidth of the noise at the output is unchanged, but the magnitude is reduced. This leads to a smoother output than the low-pass filter, due to the lower peak magnitude of the impulse response, and in particular the fact that a noise spike entering the low-pass filter is immediately (technically, with a one-sample delay) passed to the output by the low-pass filter's maximum weighting.

By comparison, the cascaded average filters create a filter with gradually ramping weightings, which ramp from zero up to a peak, then a decreasing weighting towards the end of the impulse response. This is akin to the Hanning or other raised-cosine window

functions which provide improvements over a uniform window when carrying out Fourier transforms. The result is that the noise is much better attenuated by these cascaded filters.

The graphs below show the outputs from the example filters, when the input signal is Gaussian noise with an RMS value of 1. The low-pass filter and slew-rate filters are still configured to have 5RC times of 5 cycles and slew rates of 1 over 5 cycles, respectively, which are designed to match the response of the exact-time averaging filters for the unit step scenario. Due to the nature of the data and to add clarity, the data is presented on several separate graphs with the same scales for comparison.
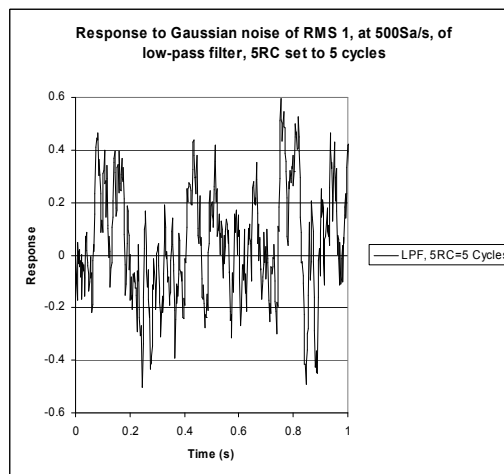
Figure 3-24 : Response to noise of the low-pass filter
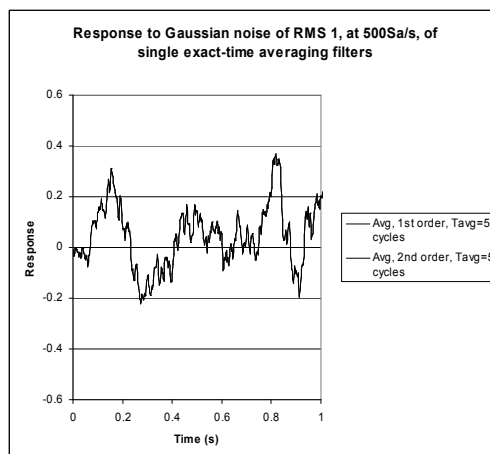
Figure 3-25 : Response to noise of the un-cascaded 1st and 2nd order exact-time averaging filters

93
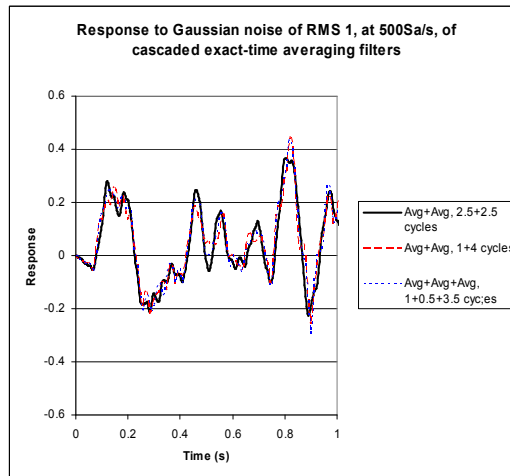
Figure 3-26 : Response to noise of the cascaded exact-time averaging filters
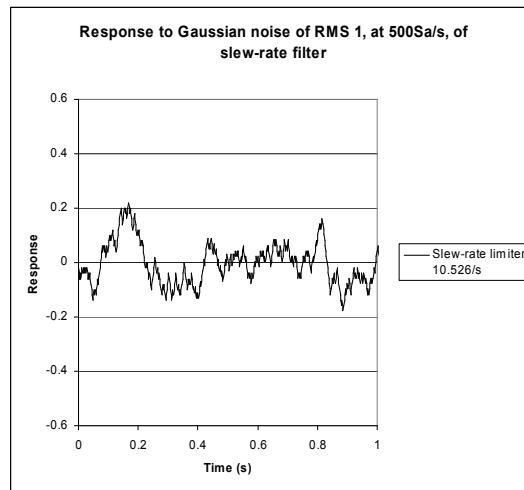
Figure 3-27 : Response to noise of the slew-rate filter

The findings from Figure 3-24 to Figure 3-26 are as expected. The averaging filter reduces the noise better than the low-pass filter. The 1st and 2nd order averaging filters have almost identical performance in noisy environments. The output from the cascaded averaging filters in Figure 3-26 are of the same overall magnitude as the un-cascaded filter shown in Figure 3-25, because the lower frequency noise components are attenuated identically. However, Figure 3-25 contains high-frequency noise which has been removed on Figure 3-26 by the superior noise attenuation of the cascaded filter.

Figure 3-27 shows the performance of the slew-rate filter for the same noise input. The slew rate filter, in this case, performs better than all the other filters.

3.3.5   Ripple rejection of filters

The final input signal analysed is a ripple of peak amplitude 1 at a known frequency. In this case this frequency is 52.632 Hz. This frequency is chosen because it is the frequency at which the exact-time averaging blocks work worst, as the interpolation errors are greatest due to having 9.5 samples per cycle at 500 Sa/s.
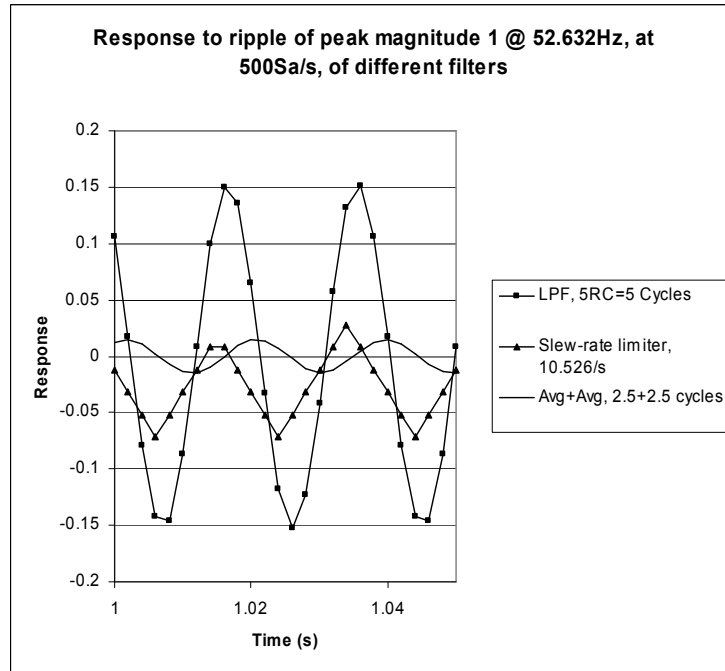
Figure 3-28 : Response to ripple of unsuitable ripple rejection filters

The filters whose outputs are shown in Figure 3-28 are not suitable for rejecting ripple. The low-pass filter has the worst response. The slew-rate limiter filter has a poor ripple, plus a DC offset appears due to the filter's non-linearity (clipping). The cascaded averaging filters with times of 2.5 cycles plus 2.5 cycles are also relatively ineffective. This is because each filter does not span a time which is an integer multiple of the ripple period.

In contrast, the filter combinations shown in Figure 3-29 are very suitable filters for ripple rejection.
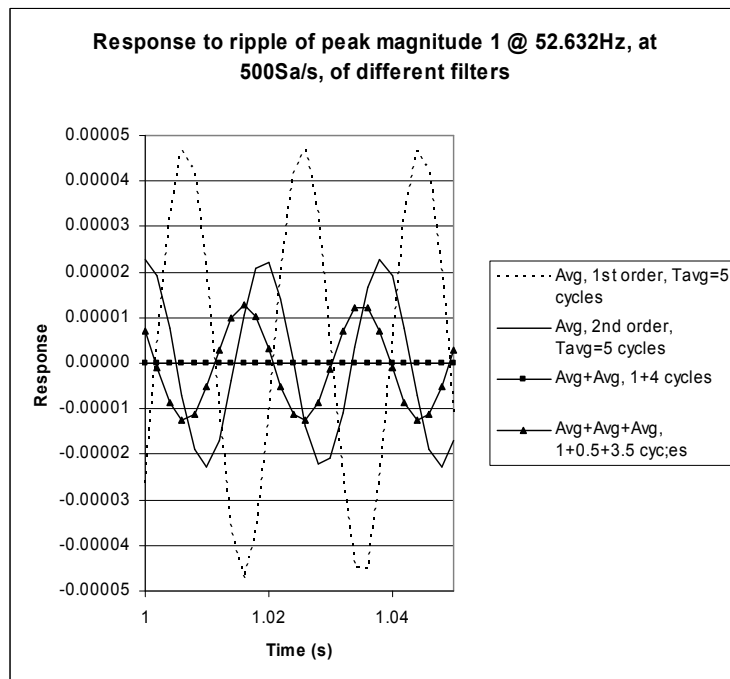
Figure 3-29 : Response to ripple of suitable ripple rejection filters

The exact-time averaging filters which use at least one section with a time average setting equal to an integer multiple of the ripple period, all provide excellent ripple rejection. The un-cascaded 1st order filter provides ≈86dB of rejection. The equivalent 2nd order filter provides ≈94dB of rejection. Cascading of multiple filters, each with time average settings equal to an integer multiple of the ripple period, provides even greater ripple rejection. The ripple is "entirely" rejected by the 1+4 cycle cascaded filter. Note that the 1+0.5+3.5 cycle filter does not perform as well as the 1+4 cycle filter, due to the non-integer cycle period timeframes used for the 2nd and 3rd averaging sections.

In reality, ripple rejection will be unlikely to be fully effective to these quoted levels, due to other noise and interfering effects which will affect both the ripple rejection itself, and also the measurement of frequency (or ripple frequency), so that the input time period for the exact-time average filter will usually be slightly in error.

3.3.6   Findings from this section

- The exact time-period averaging blocks are extremely effective at removing ripple(s) from signals if the fundamental ripple frequency is known.

- The 2nd-order exact-time averaging block removes ripple about 6dB better than the 1st-order block (a further halving of ripple magnitude). However, in the presence of noise, the advantage is lost.

96

- The exact-time-period averaging blocks out-perform low-pass filters both in their step response and noise response characteristics.

- Cascaded averaging blocks can be used both to reduce Gaussian noise at the output, and to provide further rejection of unwanted ripple at known frequencies.

- Slew-rate filters can offer very good noise reduction in some scenarios. The filter is non-linear, however, which can cause disastrous DC errors at the output for asymmetrically rippling inputs or for symmetrical input signals at certain frequencies relative to the sample rate. To counter this non-linearity the slew-rate limit can be widened but this degrades the noise rejection performance. Also, if the limiting slew-rate is too high relative to the noise or ripple, the filter becomes completely useless as noisy signals pass straight through. There are some scenarios where a slew-rate filter may be the most effective solution, but these have to be carefully examined and justified. An alternative acceleration-limiting filter was also investigated. This has appeal for a number of reasons, but this filter can oscillate under certain input conditions. Therefore it was not deemed robust enough to include in any solutions presented in this thesis.

- Low-pass filters do not provide the best solution for responses to steps, noise, or ripple. Kalman filters, being at root a type of variable low-pass filter, are thus unlikely to offer a good solution.

3.4 Analogue front-end, ADC and digital pre-processing design/considerations, including a novel DC blocking technique

The sampling hardware should incorporate hardware anti-aliasing filters before the ADC stage, to minimise the measurement errors due to aliased harmonics (see section 4.2). Gaussian noise and DC offsets will be introduced by electronic components both before and during the low-pass filtering stage (which probably also includes amplifiers, isolators etc.). The ADC stage will also introduce quantisation noise, small amounts of non-linearity, and some DC offset.

After the data has been sampled, gain and ADC channel-channel timing skew calibration corrections can be applied. Then, a sensible precaution is the inclusion of a digital high-pass filter, of some form. This filter has two potential purposes:-

- It can be used to flatten the gain vs. frequency relationship of the overall (analogue+digital) filter response around the nominal frequency point, thus minimising any gain calibration correction factors.

- It removes DC bias error components which are introduced by instrumentation

amplifiers and the ADCs. These can corrupt an all-harmonic RMS measurement since the 0th harmonic is a valid component. Also, as will be seen in section 3.11, any DC bias can also corrupt fast ½-cycle current measurements.

After careful consideration, this thesis proposes the use of a novel DC blocking filter rather than a digital high-pass filter. In the sections which follow, the analogue anti-aliasing filter is designed. Then, the calibration and digital high-pass options are considered and designed.

3.4.1   Low-pass anti-aliasing filter design

To design the low-pass filter, a trade-off is made between the rejection of aliased harmonics versus distortion of the desired signal below the Nyquist frequency. A sensible compromise is a 2nd order low-pass filter with a cut-off frequency set to 1/3rd of the Nyquist frequency. The filter is implemented as two first-order RC filters cascaded, rather than a single LC filter. This avoids the need for damping, due to the positive gain hump at resonance of an LC filter. At the cut-off frequency, the gain is -6dB, and at the Nyquist frequency the gain is -20dB (voltage amplitudes for unwanted harmonics reduced to <10% of their unfiltered voltage amplitudes. Gain reduces at 40dB/decade for higher frequencies. Scaling the filter cut-off frequency to the Nyquist frequency allows instruments with higher sample rates to process the higher order harmonics successfully, which will improve the measurements of all-harmonic RMS and THD if the instrumentation is noise-free and linear enough.

An increased filter order or decreased cut-off frequency would reject more higher-order harmonics, although attention must be paid to the group delay introduced by the filter. A 1st order low-pass filter set to a cut-off frequency of 125Hz (½ of the Nyquist frequency at 10 samples per cycle) will introduce a phase lag of 21.8° to a 50Hz input waveform. Two low-pass filters cascaded will double the lag to 43.6°, or 1/8th of a cycle. For this reason, the filter cut-off frequency should not be decreased below about 125Hz, which becomes relevant for sample rates of less than 750 samples/s, i.e. less than 15 samples per cycle at 50Hz. The resulting formula for calculating a suitable low-pass filter cut-off frequency for a nominally 50Hz system is thus:-

$$F_{cLPF} = \frac{SamplesPerCycle \times 50}{6} = \frac{F_{Nyquist}}{3} \text{ if } SamplesPerCycle \geq 15$$

$$F_{cLPF} = 125 \text{ otherwise}$$

(3.2)

By equation {B.4}, the gain of the two cascaded low-pass filters should be:

$$Gain = \frac{1}{\left(\sqrt{1+\left(\dfrac{F}{F_{cLPF}}\right)^2}\right)^2} = \frac{1}{1+\left(\dfrac{F}{F_{cLPF}}\right)^2}$$

(3.3)

The ideal phase lag through the filters can also be calculated from equation {B.4} as:

$$Phaselag = 2\arctan\left(\frac{F}{F_{cLPF}}\right)$$

(3.4)

The physical realisation of the filter may be active or passive circuits, quite likely a cascaded pair of operation amplifiers with capacitative elements in the feedback paths. Thus, the filters may introduce Gaussian noise into the signal. Also, prior to the low-pass filters is likely to be an isolation amplifier (optical or capacitative barrier). Some of these devices operate by chopping the signal at high frequency, and some of these high frequency components will appear as Gaussian noise at the filter output (and in the sampled waveforms) due to parasitic component behaviour and aliasing. These types of noise are included in the overall measurement requirements in section 2.9.

3.4.1.1   Time response of the anti-aliasing filter (and ADC)

The time response of the anti-aliasing filter can be visualised most easily by direct simulation. The test scenario is a 1pu input sinusoid at nominal frequency, which undergoes a 2 cycle brownout, beginning and ending at the peak of the cycle, so as to cause greatest disturbance to the filters. This waveform is generated at a high sample rate in simulation, to simulate analogue hardware. (An analysis of the sample rate required to accurately model an analogue filter is given in appendix B.2.2). The delay due to sampling can also be shown by using the Simulink "rate transition" block to simulate sampling. Two cases are shown here:-

- 10 samples per cycle, 2 cascaded low-pass filters with cut-off frequencies of 125Hz

- 30 samples per cycle, 2 cascaded low-pass filters with cut-off frequencies of 250Hz

At 10 samples per cycle (see Figure 3-30), the phase lag of filter is 43.6° when the input is a steady sine wave at nominal frequency. This is not a problem so long as all voltage and current inputs on all phases are processed using the same, matched (or calibrated) sets of filters so they remain coherent. Exact measurement and calibration of this phase lag might be important if the measured phases of the inputs compared to an absolute time reference

(e.g. GPS clock) are to be used to communicate phase information to other similar, distant control devices/relays. This might be relevant for a loss-of-mains or islanding detection system based upon relative phase measurements at different, distant nodes within a power system.
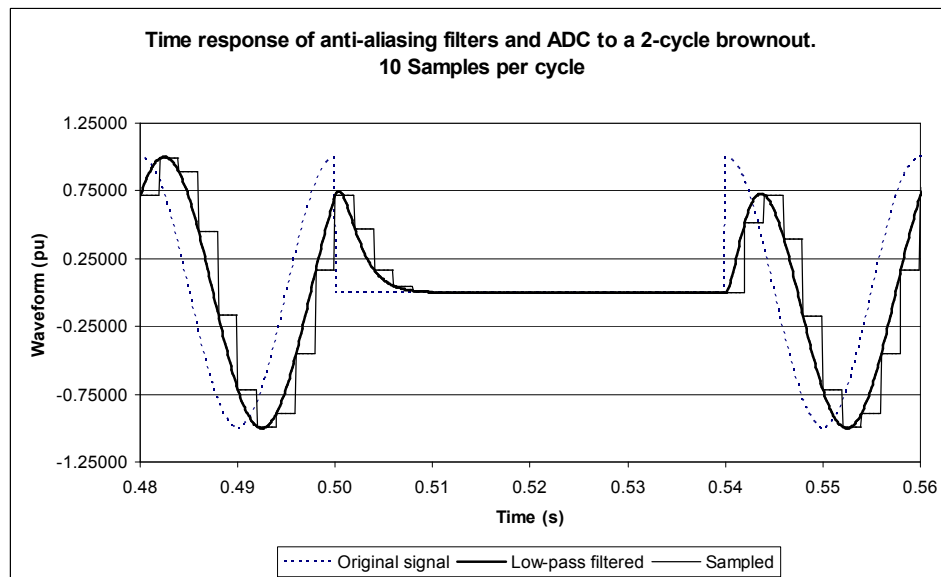
Figure 3-30 : Low-pass filter response and sampling delay at 10 samples per cycle

When a sudden perturbation occurs, the time response of the 2 low-pass filters with 125Hz cut-off frequency, plus ADC sampling, is approximately 0.01 second (½ a cycle).

A way of more theoretically calculating the "reaction time" is that the average "reaction time" of the cascaded pair of low-pass filters and sampling at some frequency F will be:-

$$\text{"SteadyStateReactionTime"} = \frac{2\arctan\left(\dfrac{F}{F_{cLPF}}\right)}{2\pi F} + \frac{Ts}{2}$$

(3.5)

Which accounts for the steady-state phase lag from equation (3.4) plus the average sampling delay (which will be half of the sample rate). For 50Hz, at 500Sa/s, with a low-pass cut-off at 125Hz, this equates to only 0.0034s (1/6th of a cycle)

During a transient, the "reaction time" appears to be larger than the steady state value calculated by this equation. This is due to the exponential decay nature of the filter after a step function input. The "worst case transient reaction time" can be re-evaluated by taking the "5RC" value of the 2 low-pass filters, which is the time taken to settle to 99.3% (1-e^{-5}) of a step function input. In this case we obtain

$$\text{``SteadyStateReactionTime''} = \frac{1.44 \times 5}{2\pi.F_{cLPF}} + \frac{Ts}{2}$$

<div align="right">(3.6)</div>

where 1.44 is the factor by which the settling time for a cascaded pair of low-pass filters takes to settle to 99.3% of the step function value, compared to a single low-pass filter. (This value found by experimental simulation). This evaluates, at 500Sa/s, with a low-pass cut-off at 125Hz, to 0.010s (½ a cycle), and matches the estimated value from the simulation in Figure 3-30. This result is quite conservative, and accounts for a complete settling of the filter to a transient input. The filter "reaction time" must be borne in mind when accounting for the total latency of the measurement system, which will be made up of this time plus the digital processing/averaging time. The total time will be of most relevance where fast-acting relay action is required.

At 30 samples per cycle, the response time of the filters decreases below 0.005 second (¼ cycle), as shown in Figure 3-31. This is due both to the higher cut-off frequency of the analogue filters, plus the reduced sampling delay. At higher sample rates, the latency decreases further towards zero.
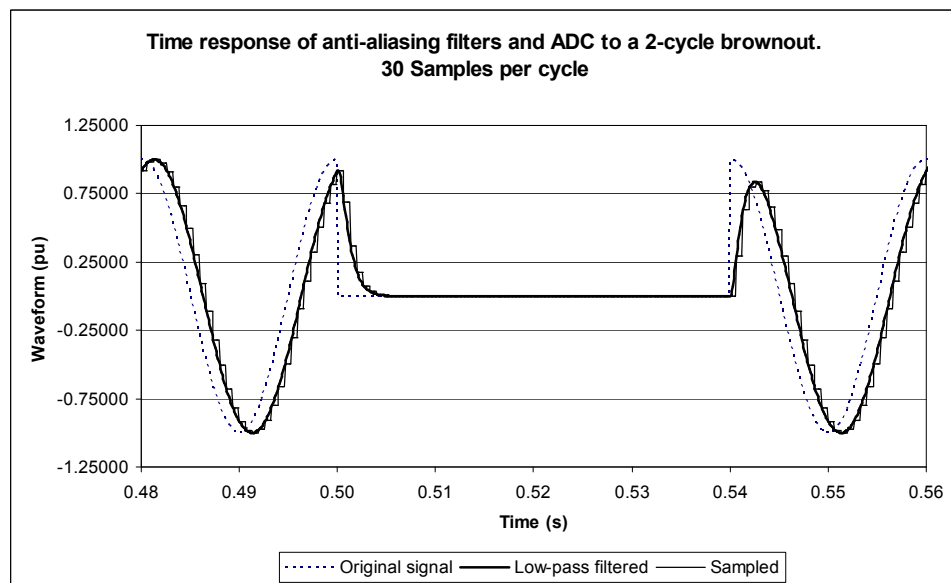
Figure 3-31 : Low-pass filter response and sampling delay at 30 samples per cycle

3.4.2   ADC effects

The ADC introduces quantisation noise. Typically an ADC has 12 bits. Modelling this in a theoretical form is difficult, but introducing the effect to a simulation is relatively easy. For voltage measurements, the scaling can be set so that, for example, 2pu +ve or –ve peak values cause 0 or 0xFFF (2^{12}-1) full-scale readings on the ADC. This means that the

nominal -1pu to +1pu voltage range is divided into 2^{11} discretised values, and so the introduced errors are very small. For current measurements, the maximum measurable current (without saturation of the CTs, instrumentation, or ADCs) must be decided, and the ADC scaling set from there. ADC quantisation can therefore be significant if the actual current flowing is small relative to the peak measurable current. A real ADC also has slight non-linearities, which are not addressed in this thesis, (nor are VT and CT saturation/linearities), since this analysis is focussed on the errors due to digital processing. The ADC non-linearity is also small compared to the potential maximum harmonic content of the input signals (section 2.7.2). In addition to the theoretical quantisation noise, often the noise within the ADC hardware actually makes the lowest bit(s) of the ADC random. Allowance for this is made in the measurement requirements in section 2.9.

3.4.3   Post-ADC calibrations/corrections

After the signal has been sampled, there are several tasks to perform before the Fourier analysis is carried out:-

1. Removal of DC bias components which are introduced by instrumentation amplifiers and the ADCs.

2. Correct the input signals for relative phase offsets, due to ADC channel-channel time skews (if the ADCs are multiplexed) or different VT/CT performances.

3. Amplitude calibrations for each measurement channel and (calculation of) the overall phase calibration of the input signals. The calibration coefficients can be based upon manual "one-time" measurements at just one (nominal) frequency or at several frequencies. This corrects for the gains of all hardware including VTs, filters, and ADC etc. It might be sufficient to use a single frequency if the gain of the VTs is flat enough, and if the filters are manufactured to a tight enough tolerance. If this is not the case, then several frequencies may need to be calibrated and interpolation used between the calibrated frequencies.

4. Amplitude and (calculation of) phase correction for off-nominal frequency inputs to correct for known gain & phase response transfer functions of the anti-aliasing filters and any post-ADC digital filters (such as the DC block).

The calibration values for steps 2 & 3 can be measured and/or deduced from specifications at or before installation, and stored in a table within the measurement computer system. The calibration values for step 4 can be deduced by inverting equations (3.3) and (3.4), and by using similar techniques for any digital post-ADC filters.

The application of the gain corrections from steps 3 and 4 is straightforward since this involves purely a multiplication of the sampled signal value by the calibration value. The application of the phase corrections requires substantially more care.

The method of application, particularly the ADC time-skews, varies depending upon the way that the sampled waveforms will be analysed. It can be achieved in two ways:-

1. by delaying the input signals appropriately and interpolating between samples so that each input signal then appears to be coherent (2nd order interpolation has been shown to work well). This is definitely the appropriate way to deal with the small ADC channel-channel time-skews within inverter control systems which convert 3-phase sampled data directly into the dq frame without a Fourier analysis stage.

2. by carrying out Fourier analysis of each sampled channel directly, and then applying a post-correction to the measured phase of the data.

During this thesis, both methods have been used. The additional factors which influence the choice of method include the computational effort required for subsequent algorithms such as the processing of Fourier measurements of 3-phase signals into the positive and negative sequence components. When all things are considered, the best method is to use both techniques together. Firstly, the small relative time skews between all voltage and current measurements (ADC channels) at a single node should be corrected up front via the first method. This allows calculation right through to the sequence analysis with minimal trigonometric calculations. The overall larger common (absolute) time skew for all these channels (relative to some known/fixed reference) should be corrected at the end, after all magnitudes/phases and sequence analysis is complete.

3.4.4   DC block / high-pass filter design

Immediately after the ADC (and before application of any calibrations) a DC block is desirable. This removes DC bias error components which are introduced by instrumentation amplifiers and the ADCs. Two possible options are compared in the following sections. The first is a digital 1st-order high-pass filter. The second is a novel DC blocking algorithm based upon the exact-time averaging technique previously introduced.

3.4.4.1   Digital high-pass filter option

A high-pass filter blocks the DC component but can also be used to flatten the gain of the entire cascaded filter section (low-pass + digital high-pass) vs. frequency around the nominal frequency point. Appendix B.3 and equation {B.9} show how the required cutoff

frequency of the high-pass filter F_{cHPF} can be calculated to achieve this goal at nominal frequency f, when cascaded with two low-pass filters with cutoff frequencies of F_{cLPF}.

For 10 samples per cycle (500 Sa/s), with the 2 low-pass filters set at 125 Hz (½ Nyquist) and f=50 Hz, this results in a high-pass filter cut-off frequency of 30.86Hz and an overall response shown below (normalised to 0dB @ 50Hz).
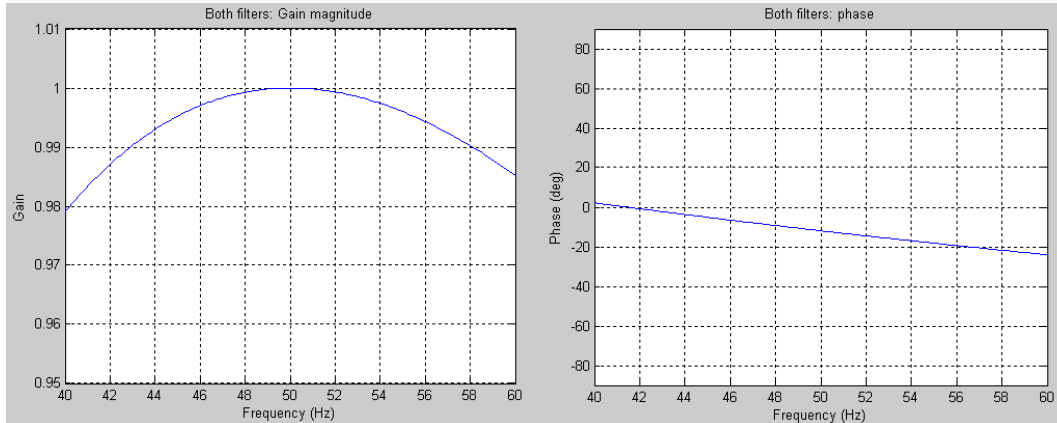
Figure 3-32 : Low-pass / High-pass filter combination for 500 Sa/s

Note that the overall steady-state phase response has been "improved" from a 44° lag at 50Hz (due to the low-pass filters alone, see section 3.4.1) to only a 10° lag. However, the actual response of the system to sudden changes in input will still be lagged in time by approximately the original time lag group delay of the low-pass filters (1/8th of a cycle) plus the ADC sampling lag, plus the group delay of the digital high-pass filter.

At higher sampling frequencies, the low-pass filter cut-off frequency can be increased and the high-pass cut-off frequency required for the flat gain condition decreases. The flatness of the gain curve improves. For example, at 30 samples per cycle (1500 Sa/s at 50Hz), the low-pass cut-off is 250Hz and the high-pass cut-off is 14.43Hz. The resulting filter response is as below:
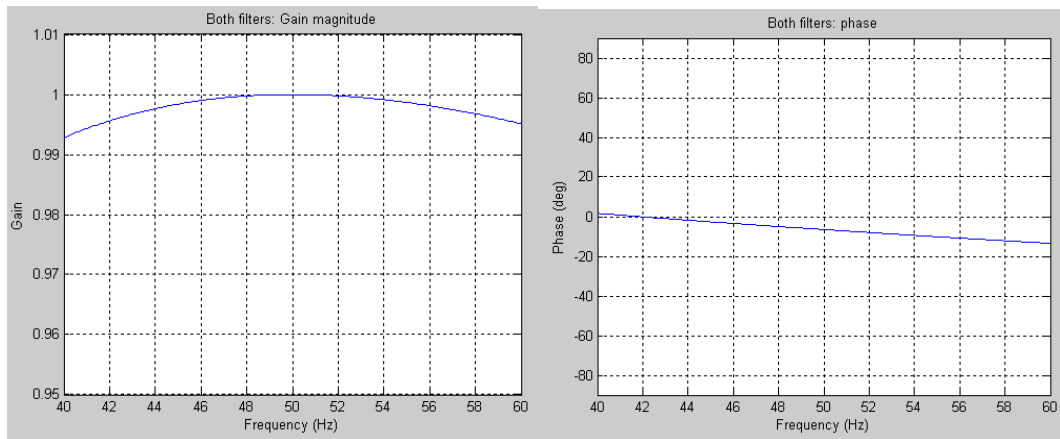
Figure 3-33 : Low-pass / High-pass filter combination for 1500 Sa/s

3.4.4.2   Novel DC blocking technique option

An alternative to the digital high-pass filter is a novel DC block, designed using the blocks described in section 3.2. The idea is to measure the amount of DC present on the input signal, and then subtract this from the input signal. This means that the DC blocking filter has absolutely zero propagation delay for AC signals. The rationale for using this block instead of a standard high-pass block is:-

1. That the exact-time averaging blocks can be used to reject "ripple", orders of magnitude better than the high-pass filter. Since the input signal is expected to be a sine wave, the entire input signal is in fact "ripple".

2. That the DC offset of the sampled data is expected to be relatively constant, due to component behaviour within the instrumentation (mainly isolation amplifier offsets and operational-amplifier offsets). Thus, the measured DC offset term can be smoothed using a slew-rate filter with a low maximum slew rate setting. Such a slow slew-rate filter rejects noise extremely well, as shown in section 3.3.4.

3. That, although flattening the overall filter gain at nominal frequency by using a high-pass filter is desirable, it is not essential since the low-pass filter gain slope can be corrected during calibration as described in section 3.4.3.

The design of the DC blocking filter is shown below:-
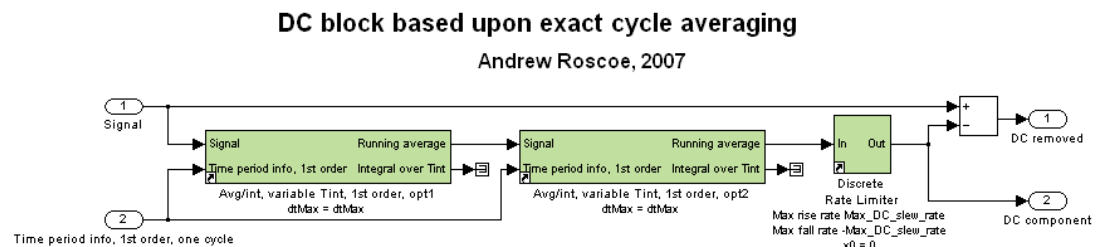
Figure 3-34 : DC block

This block averages the input signal over 2 cascaded cycles. This produces a DC output with virtually zero interpolation ripple (see section 3.3.5) for an input signal which is made up of a fundamental plus harmonics, assuming the measurement of the signal frequency is correct. A departure from these assumptions, such as noise, inter-harmonics, sub-harmonics, or inaccurate frequency estimation, results in an averaged, bandwidth-limited but slightly rippling signal from the two cascaded average filters. Next, a slew rate filter can safely be used in this application to further reduce the effects of noise. The slew rate limit could be reduced to very low levels <<0.1 pu/s, since the DC offset of the instrumentation will be almost static. However, to speed up settling of the filter both in reality and in simulation, a value of 0.1 pu/s for the slew-rate limit is a sensible compromise. This setting also influences the behaviour of the block during faults which might exhibit a decaying DC component on the measured waveforms (voltage or current). By setting the slew rate to 0.1 pu/s, the DC component will initially be passed straight through the DC block to further processing, but after 1 second the block will filter out 0.1 pu of the DC component (if the DC component is still >0.1 pu). If it is desired to measure DC components during faults with high accuracy, but still remove DC bias due to instrumentation, then the slew rate should be set <<0.1 pu/s. On the other hand, if removal of the bulk of DC components even during faults is desired, then the slew rate should be set to >>0.1 pu/s.

To illustrate the benefit of this filter over the high-pass filter described in 3.4.4.1, a simple Simulink simulation was created. This operates at 500Sa/s, 10Sa/cycle @ 50Hz. The input is a synthesised sine wave of 1pu peak amplitude at 52.6316Hz into a high-pass filter and the DC blocking filter. This frequency is the worst frequency for interpolation ripple within the averaging blocks as there are 9.5 samples per cycle. The high-pass filter cut-off frequency is 30.86Hz (see section 3.4.4.1). A large DC offset (0.25pu) is applied to the signal. The DC blocking filter thus takes 0.25 seconds to initially settle. At t=5s, a hard fault is simulated, and the fault is removed at t=5.1s.

Figure 3-35 shows the response of the two filters at the instigation of the fault. On this graph, the input signal has been adjusted downwards by the DC bias of 0.25pu for the purposes of plotting, to create a reference (correct) value. Clearly, the DC blocking filter tracks the reference signal exactly, as the traces are indistinguishable. The high-pass filter, on the other hand introduces small lags in the signal during normal operation, and also causes lag and decay effects during fault conditions.
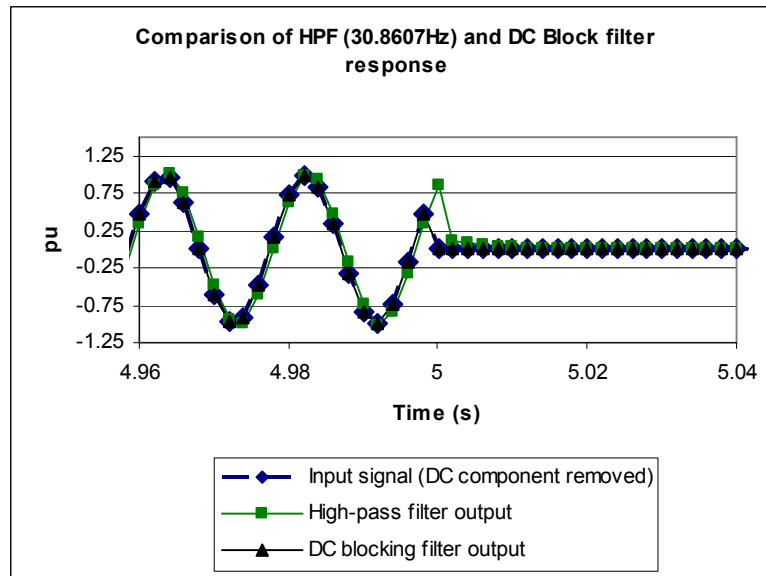
Figure 3-35 : DC block vs. high-pass filter performance

Figure 3-36 shows the errors from the two filter types, which are deduced by subtracting the reference signal value from the filter outputs. The DC Block has a peak error of <0.005pu, whereas the high-pass filter has a peak error of almost 1pu, due to lag in the filter.
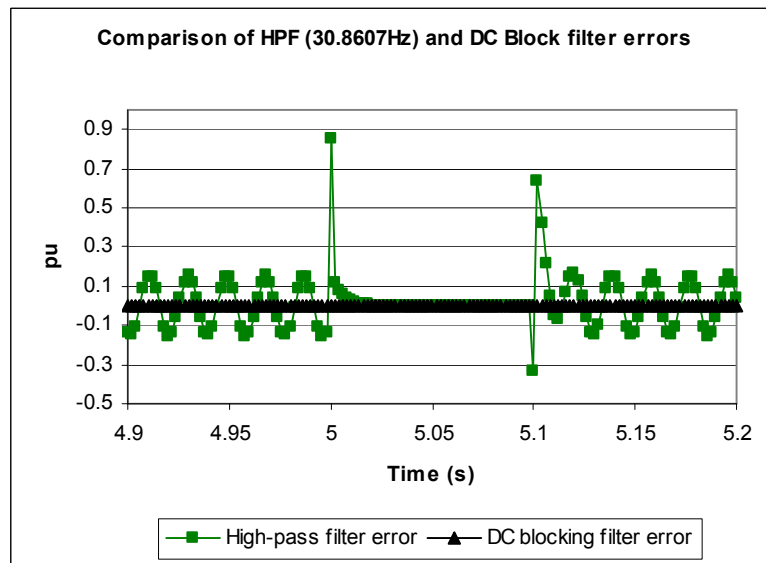
Figure 3-36 : DC block vs. high-pass filter errors

3.4.5   Findings from this section

- A cascade of 2 1st-order low-pass filters, set to 1/3rd the Nyquist frequency, or a minimum of 125Hz, is a sensible, simple filter combination to remove higher-order harmonics which would otherwise be aliased during the sampling process.

- The group delay and latency of the low-pass filters and ADC sampling delays/smears transient response by about 1/6th cycle (steady state) to ½ cycle (transient) at 10 samples per cycle, and a 1/12th cycle (steady state) to ¼ cycle (transient) at 30 samples per cycle.

- The known low-pass filter characteristics can be used to calculate correction factors (amplitude and phase) for waveforms measured at off-nominal frequencies.

- The required calibrations for amplitude and phase can be introduced at sensible points in the digital processing.

- A novel DC block, built using a cascade of 2 exact-time averages and a slew-rate limiter, provides a much better way of removing unwanted instrumentation DC bias from signals, than a digital high-pass filter.

3.5   SimPowerSystems Fourier and RMS measurement blocks

This section introduces the "Discrete Fourier" and "Discrete RMS value" algorithms which are part of the MATLAB SimPowerSystems blockset within Simulink. These are relevant because they are the starting point for the more advanced methods for measurement of amplitude and phase measurement which are subsequently developed and analysed in this thesis.

The SimPowerSystems blocks are shown in Figure 3-37 and Figure 3-38. They measure a single-phase signal amplitude (and phase in the Fourier case), given a fixed estimate of the signal fundamental frequency. The Fourier block can measure the amplitude/phase of any harmonic component by adjusting an input parameter n at compile-time. The multiplication factor k is usually 2, but set to 1 for measurement of the 0th harmonic.

The algorithm evaluates the expression

$$F = \frac{k}{T}\left[\int_{t_0-T}^{t_0} y(t)\cdot\sin(\phi)\cdot dt + j\int_{t_0-T}^{t_0} y(t)\cdot\cos(\phi)\cdot dt\right]$$

where $\phi = 2\pi\cdot nf\cdot t$, f is the estimate of frequency, t_o is "now" and T is the integration time ($1/f$ for a single-cycle measurement).

In the case of the fundamental measurement (with n=1), the magnitude is then given by $\theta = |F|$ and the signal phase (relative to the correlating waveform) is given by $\theta = \angle F$.

Notably, the "absolute phase" is then given by $(\theta + \phi)$ and the fundamental may be estimated by $|F| \cdot \sin(\theta + \phi)$. A packet of data containing the phase $(\theta + \phi)$, the frequency f and an accurate timestamp (e.g. from a Global Positioning system) can be passed to distant protection/control systems. Upon receipt, the phase data can be compared to other similar data accurately, accounting for variable latencies in the communications channels.
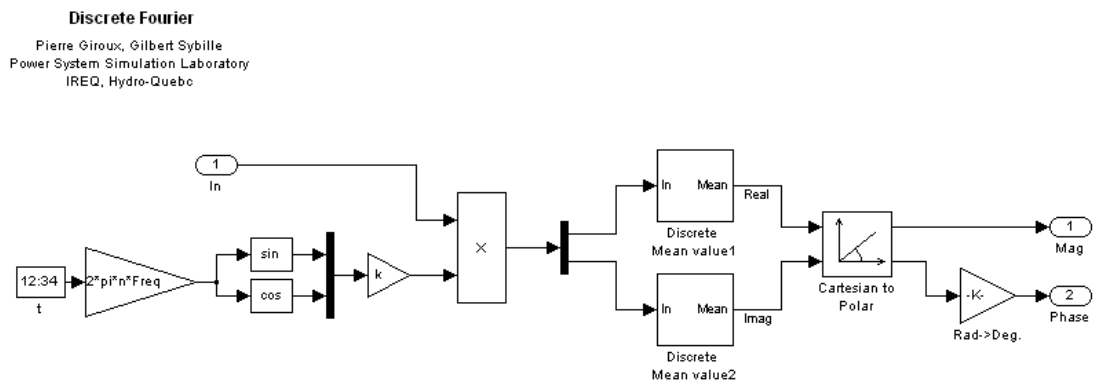
Figure 3-37 : SimPowerSystems "Discrete Fourier" block
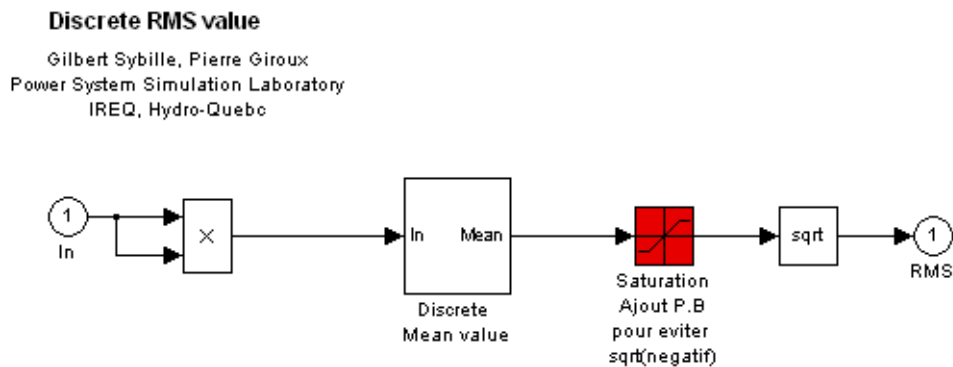
Figure 3-38 : SimPowerSystems "Discrete RMS value" block

3.6 Development of simplest Fourier and RMS amplitude measurement block (1st and 2nd order integration & interpolation)

The obvious improvement to these blocks is to make the parameter *frequency* a dynamic input rather than one that has to be fixed at compile-time. Due to work in the previous sections, this can now easily be done by substituting the new exact-time averaging/integration blocks developed in section 3.2. These new blocks not only allow dynamic setting of the *frequency* parameter, but also add the computational robustness required for real-time safety-critical deployment (which is not present in the existing SimPowerSystems blocks).

Considering that many such Fourier and RMS measurement blocks may use the same estimate of signal frequency, it makes sense to bring out some of the calculations into pre-calculations which can be used for many Fourier and/or RMS blocks. This is similar to the pre-calculation methods of section 3.2.3; in fact the *TimePeriodInfo* pre-calculation of Figure 3-5 is embedded within the Fourier pre-calculations, because the Fourier correlation time period needs to be averaged/integrated using the exact-time period averaging blocks. Of particular benefit is the pre-calculation of the trigonometric functions sine and cosine, since these are relatively expensive in terms of CPU time.

The two key building blocks for the algorithms using 1st order integration/interpolation are shown below. The 2nd order versions are almost identical; only the averaging blocks being implemented differently as described in section 3.2.2.
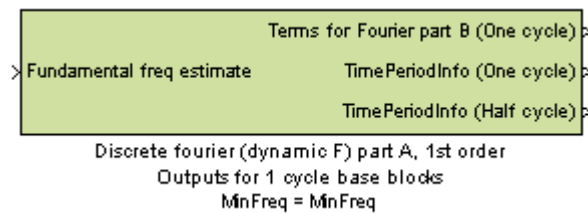
Figure 3-39 : Pre-calculations (part A) for Fourier analysis block (overview)
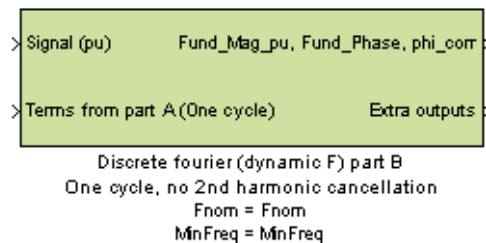
Figure 3-40 : Simplest Fourier analysis block (part B) (overview)

Inside the pre-calculation block (part A), there are no surprises. The reader may for now ignore the "half-cycle" output, which is described in later sections.
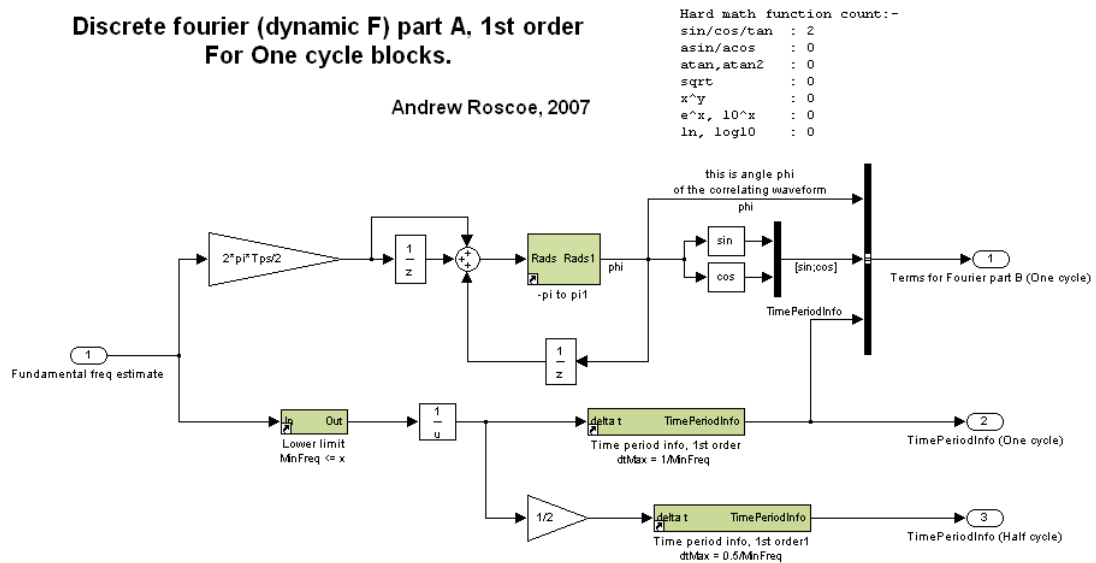
Figure 3-41 : Pre-calculations (part A) for Fourier analysis block (detail)

Similarly, inside the Fourier calculation of Figure 3-42, (here called "part B"), the core code is familiar and closely resembles Figure 3-37. The differences are:-

- The averaging blocks are replaced by the better, newly developed blocks (see section 3.2).

- There are references to "no cancellation". The reader can ignore these for now, as they discriminate this algorithm from a more complex algorithm explored in section 3.8.

- There is code which detects rapid changes of amplitude. This is done by using a two-sample differentiator block. This is done so as to give a fast warning of transient conditions.

- There are several additional outputs from the block, such as the transient detection etc. These are used for debug purposes, algorithms using 2nd harmonic cancellation (see section 3.8), and also frequency measurement algorithms (see section 5.4.2).

- The meaning of the "phase" output is clarified. Unmodified, this output gives the phase of the measured signal, relative to the phase of the pre-calculated sine/cosine correlation waveforms in the "part A" pre-calculated data. Thus, the phase outputs of any similar Fourier blocks using the same "part A" data can be

compared directly together in a relative manner. Also, if the phase output *Fund_phase_rel_phi_corr* is added to the correlation waveform phase *phi_corr*, then Φ_{abs}, the absolute phase of the input signal relative to a positive-going zero crossing, is determined. This value can be used to recreate an estimate of the signal fundamental via $\sin(\Phi_{abs})$ times the measured amplitude. Φ_{abs} can also be extremely useful to pass between remote systems to compare phases between AC waveforms at different locations on a network, being measured by different instrumentation systems which have unsynchronised CPU clocking crystals. This information must, however, be qualified by an accurate timestamp (from the GPS system, for instance) and also the estimate of frequency. In this way, communication delays can be backed out of the data to enable an exact comparison of relative phases to be made.

Discrete Fourier analysis, with a single cycle base

Andrew Roscoe, 2007

```
Hard math function count:-
sin/cos/tan : 0
asin/acos   : 0
atan,atan2  : 1 (Cart to polar)
sqrt        : 1 (Cart to polar)
x^y         : 0
e^x, 10^x   : 0
ln, log10   : 0
```
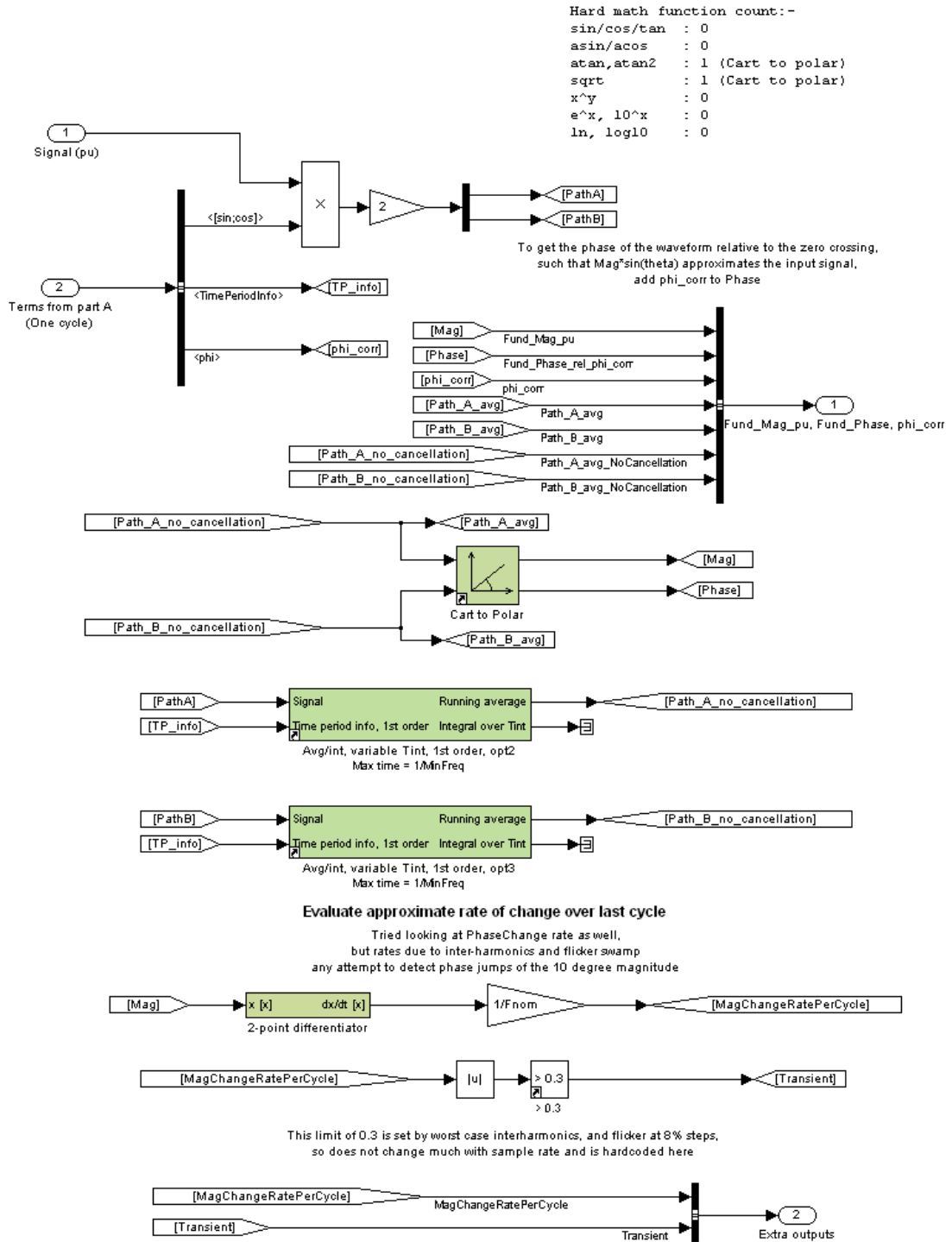

Figure 3-42 - Fourier analysis, "part B", detail

The fundamental magnitudes measured by the Fourier block of Figure 3-42 are output as peak amplitude values. These can be converted to RMS amplitude values by dividing by $\sqrt{2}$.

The all-harmonic RMS measurement block, by comparison, is much simpler. It is identical to the original SimPowerSystems block of Figure 3-38, except that the new averaging algorithms are substituted, using either 1st or 2nd order integration/interpolation.
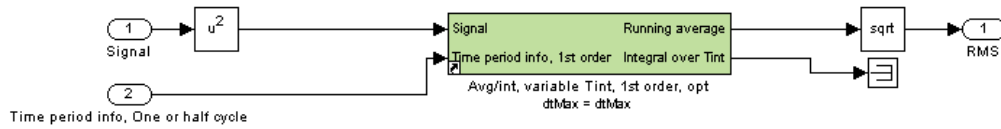
Figure 3-43 : All-harmonic RMS calculation evaluation

The algorithm evaluates the expression

$$RMS = \sqrt{\frac{1}{T} \int_{t_0 - T}^{t_0} y^2(t) \cdot dt}$$

(3.7)

where t_o is "now" and T is the integration time ($1/f$ for a single-cycle measurement).

THD can be calculated from the RMS amplitude of the fundamental (V_1) and the all-harmonic RMS amplitude V_{All}, by the following relationship

$$THD(\%) = 100 \times \frac{\sqrt{V_{All}^2 - V_1^2}}{V_1}$$

(3.8)

where the RMS amplitude of the harmonic content V_h (everything except the fundamental, i.e. DC bias plus 2nd, 3rd, 4th, and all higher harmonics) is

$$V_h = \sqrt{V_{All}^2 - V_1^2}$$

(3.9)

In cases where the input waveform has already been passed through a DC blocking stage, the measurement of V_{All}, THD, and V_h will not include the DC bias (0th harmonic).

3.7 Amplitude measurement errors due to integration and interpolation at low sample rates, using pure sinusoid inputs

The 1st and 2nd order algorithms described in section 3.6 produce perfect results if the input is a perfect sinusoid at nominal frequency, so that the number of samples per cycle is an integer number. This is true even when the samples do not fall at the zero crossings, as the interpolation errors at the beginning and end of each measurement timeframe cancel each other out. However, when the actual frequency does not result in a period which equals an integer number of sample times, the measurements exhibit integration/ interpolation errors which show up as ripples. The worst input frequencies have been found to be those which result in N±½ samples per cycle, where N is an integer. For example, with nominal settings of F_{nom} Hz and N samples per cycle, the worst expected input frequencies can be calculated by

$$Fworst = \frac{F_{nom}N}{\left(N \pm \frac{m}{2}\right)}$$ where m is any sensible odd integer to give a positive frequency

(3.10)

So, for 50 Hz and 10 samples per cycle, the worst frequencies closest to 50 Hz would be 47.619 and 52.632 Hz (10.5 and 9.5 samples per cycle). At these frequencies, it does not matter what the phase of the incoming signal is relative to the sampling points, the ripple errors are always of the same magnitude. For example, the start (zero crossing) of a cycle at 52.626Hz may fall exactly on a sample point. In this case the end (next zero crossing) of the cycle will fall exactly between two sample points, resulting in an interpolation error. Conversely, if the start (zero crossing) of a cycle at 52.626Hz falls exactly between two sample points, then the end (next zero crossing) of the cycle will fall exactly on a sample point. This results in the same interpolation error. Phases in between these two examples result in the same interpolation error magnitude.

However, due to the effects of lowering the number of samples per cycle at higher frequencies, the interpolation error amplitude will be larger at, say, 52.632 Hz than 47.619 Hz, simply because there are less samples per cycle at 52.632 Hz and the interpolation takes place over longer timeframes. Thus, the actual performance vs. frequency will be a combination of the predictions of equation (3.10), plus a tendency for ever higher errors at ever higher input frequencies (and in the presence of higher-order harmonics).

To evaluate the actual measurement errors of the single-cycle Fourier measurement blocks due to integration and interpolation, a Simulink model was created. This is a multi-rate simulation designed to tests all aspects of the algorithms described above, by synthesising input waveforms at desired sample frequencies with variable amounts of Gaussian noise and harmonic distortion. Allowance is also made to simulate the effects of analogue and digital filtering, before and after the ADC stage, plus the effects of ADC discretisation. These effects are all described in section 3.4.

This model can be executed repeatedly from a pair of MATLAB scripts which first run multiple instances of the simulation across a wide range of scenarios to create a data file, and then plot the results. Using this setup, the graph below shows the maximum interpolation errors against input frequency, when using 10 samples per cycle (50 Hz nominal frequency, 500 Sa/s) and inputs of pure sinusoids between 45 and 55 Hz. In this set of simulations, the anti-alias filter response, Gaussian noise, ADC quantisation effects and DC bias/block are not modelled so as to focus purely on the algorithm integration/interpolation error. The Simulink model uses the Fourier blocks from section 3.6 (both 1st and 2nd order versions), correlating the input waveform over exactly one cycle. The results are shown in Figure 3-44.

Figure 3-44 : Fourier analysis of fundamental. RMS errors due to integration & interpolation @ 10 Sa/cycle. 1st order (solid) and 2nd order (red dashes) methods.

The error magnitudes shown in Figure 3-44 (and following) plots are the RMS of the instantaneous ripple error values, with the mean of the "root mean squared" evaluated over one cycle.

Note that the 2nd order methods reduce the errors compared to the 1st order methods, but that the improvement is no better than a factor of 2 at any point.

It is important to note that in this, and all other simulations in this chapter, it is assumed that the signal frequency is known. Thus, most error analyses shown in this chapter do not account for errors in the frequency measurement. Of course, if the frequency is not known then additional errors will be present. This error will be largest when a "dead" power system is initially connected via a breaker to a "live" power system. In this case, several cycles may elapse before the frequency measurement is accurate. During this time the amplitude and phase measurements will also be in error. These errors are examined in section 4.1. During brief voltage dips, the amplitude/phase measurement error can be minimised by implementing "ride-through" capability into the frequency measurement algorithm. This is specified in section 2.7.5 and implemented in section 5.4.2.

Exploring a range values of samples-per-cycle from 10 to 30, and finding the worst case errors obtained for any input frequency between 45 and 55 Hz results in Figure 3-45. Note that as the number of samples per cycle increases, the frequencies at which the errors peak occur get closer to 50Hz (and hence more likely to be observed), but that the magnitude of the errors decreases (compare Figure 3-44 and Figure 3-48).

It was anticipated that the benefit of the 2nd order methods would decrease as the number of samples per cycle was increased, even for a pure sinusoidal input. However, this proves not to be the case, on a proportionate basis. Upon further examination, it has been found that the 2nd order methods do not work at their best at the low sample rates near 10 Sa/cycle. This is because the integrations within the Fourier and all-harmonic RMS blocks end up integrating sin^2 or cos^2 type functions which have a frequency of twice nominal (100 Hz). There are only 5 Sa/cycle in these waveforms, and the 2nd order curve fitting errors are relatively large. The advantage of the 2nd order methods over the 1st order actually increases (proportionately) as the number of samples per cycle increases.

Figure 3-45 : Fourier analysis of fundamental. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods.

117
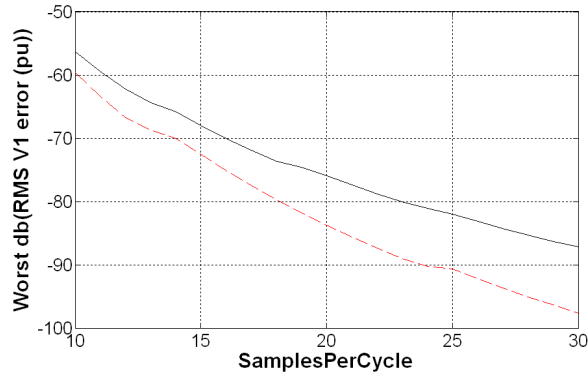
Figure 3-46 : Fourier analysis of fundamental. Largest dB(RMS errors) due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods.

The slope for the 1st order method error is approximately -56dB to -87dB for a 3-fold increase in sample rate. This is a factor of 35x, which means that the error magnitude follows approximately an $N^{-3.2}$ relationship where $3.2=\ln(35)/\ln(3)=3.2$. For the 2nd order method, the relationship is approximately $N^{-3.9}$

The ripple frequency of the error is always at twice the input frequency (shown by Figure 3-47). Thus turns out to be an exceptionally useful property and is exploited fully during the rest of this thesis (see sections 3.3.5 & 3.9 for example). The exception, for a pure sinusoid input, is when an integer number of samples spans on cycle period. In this case, the interpolation & integration error is zero and the ripple frequency is also zero (undefined). In the case of Figure 3-47, with 500 Sa/s sampling, this occurs for a 50 Hz input signal.
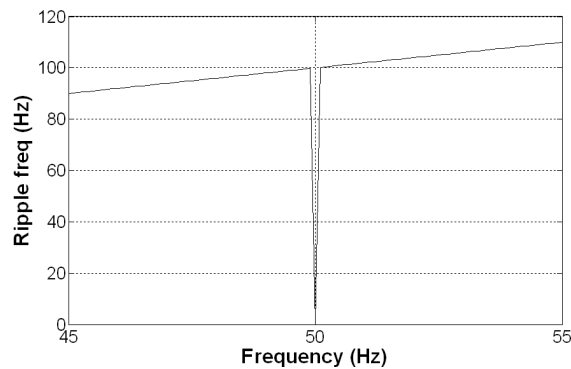
Figure 3-47 : Fourier analysis of fundamental. Ripple frequencies of largest errors due to integration & interpolation over the 45-55Hz range.

To further indicate the reduction of error magnitude and the increasing relative effectiveness of the 2nd order method, Figure 3-48 below shows the analysis technique of

118

Figure 3-44 repeated, but with sample rate increased from 500 Sa/s (10 samples per cycle) to 1500 Sa/s (30 samples per cycle). The worst error magnitudes for the 1st order method have been reduced from 0.0015 pu to 0.00004 pu, i.e. by a factor of 40. The worst error magnitudes for the 2nd order method have been reduced from 0.001 pu to 0.000012 pu, i.e. by a factor of 80.
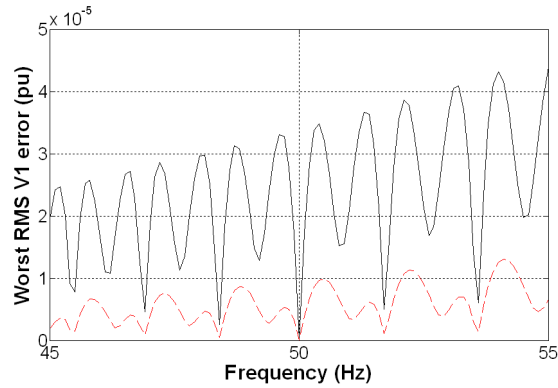
Figure 3-48 : Fourier analysis of fundamental. RMS errors due to integration & interpolation @ 30 Sa/cycle. 1st order (solid) and 2nd order (red dashes) methods.

The reductions in error magnitude versus sample rate for the all-harmonic measurements (and hence THD measurements) behave in a similar fashion to the error magnitudes for the Fourier fundamental measurements, as shown in Figure 3-49 and Figure 3-50.

Figure 3-49 : All-harmonic analysis. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods.
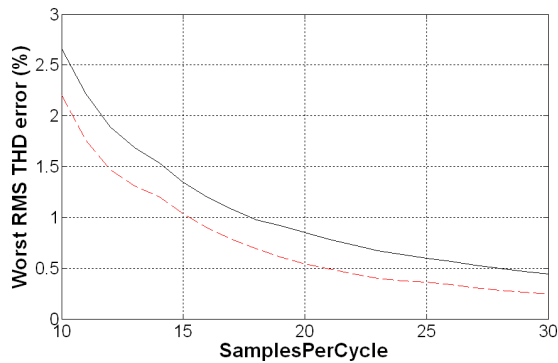
Figure 3-50 : THD analysis. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods.

The THD measurement is extremely susceptible to errors in the measurements of the all-harmonic or fundamental RMS values. THD is calculated in the model above by equation (3.8). The effect of the measurement errors on the THD calculation can be demonstrated by imagining a 0.0001pu error in the estimation of V_{All}. This might produce a THD error of 100*√(1.0001^2-1)/1=1.4%. This is a significant error in the THD measurement, considering that the BS EN 50160 specification is for 8% THD. The THD measurements made using the low sample rates in this document should be used as a guide rather than a measure. For an accurate measure of THD, a much higher sample rate needs to be used. In the context of the work in this thesis, this is useful to know but does not present a problem, since the aim is to produce algorithms for protection and control, not for power quality measures. The measurement of THD is thus not of primary concern.

3.7.1 Findings from this section (considering pure sinusoid inputs only)

- The extension to second order reduces the magnitude of the errors. At 10Sa/cycle the errors are reduced to about 66% of the errors from the 1st order methods. At 30Sa/cycle the error reduction improves to 30% of the 1st order methods.

- For any sampling rate of 10Sa/cycle or more, the Fourier and all-harmonic errors due to integration/interpolation using the 2nd order methods are less than 0.002pu. At 16 Sa/cycle, the largest error is about 0.0004pu, although the resulting THD error might still be as high as 1.25%.

- Although the Fourier and all-harmonic RMS errors are very small for pure sinusoid inputs, even at 10Sa/cycle, the 2nd order methods initially appear to be worth using since the THD measurement error is improved significantly from 2.7% to 2.2%. However, it is shown later that second harmonic cancellation can reduce the interpolation errors in the initial single-cycle Fourier correlation by much larger factors (see section 3.8). Also, again shown later, post-averaging stages using

120

cascaded 1st-order averaging filters can make the advantage of the 2nd order techniques redundant (see section 3.9). Also, the relatively small size of the integration/interpolation errors (for both 1st and 2nd order methods) and the additional error effects due to noise/harmonics (see section 3.13) mean that the 2nd order methods do not add much overall benefit given the additional processing overhead relative to the 1st-order methods.

- Referring to Figure 3-47, it can be observed that the ripple frequencies of the errors from a single-cycle Fourier transformation block, due to integration and interpolation, appear at twice the input frequency. This turns out to be a very useful and predictable property, and is used to good effect in section 3.9 which subsequently forms the basis of excellent measurement algorithms.

3.8 Addition of active 2nd harmonic cancellation

Referring to Figure 3-47, it can be observed that the ripple frequencies of the errors from a single-cycle Fourier transformation block, due to integration and interpolation, appear at twice the input frequency. The reason for this can be explained by referring to Figure 3-42, which shows the Fourier correlation. The correlation is made by taking a reference wave at the supposed frequency f, and then correlating $\sin(2\pi f)$ and $\cos(2\pi f)$ against the actual waveform which, for a pure sinusoid signal, is of the form $V\sin(2\pi f + \Phi)$ where Φ might be any number.

This gives the following correlations

$$V.\sin(2\pi f + \phi) \cdot \sin(2\pi f) = \frac{\cos((2\pi f + \phi) - 2\pi f) - \cos((2\pi f + \phi) + 2\pi f)}{2} = \frac{\cos(\phi) - \cos(4\pi f + \phi)}{2}$$

$$V.\sin(2\pi f + \phi) \cdot \cos(2\pi f) = \frac{\sin((2\pi f + \phi) + 2\pi f) + \sin((2\pi f + \phi) - 2\pi f)}{2} = \frac{\sin(4\pi f + \phi) + \sin(\phi)}{2}$$

(3.11)

The correlations therefore have a DC term plus a 2nd harmonic term. Fourier analysis needs to average/integrate these values over (traditionally) a full cycle. Over this timeframe, the second harmonic term disappears if the interpolation errors are small. However, at low sample rates the interpolation within the exact time-frame averaging has to cope with a signal at twice fundamental. For a system working at 10 samples per cycle, the 2nd harmonic has thus only 5 samples per cycle, resulting in the interpolation errors shown on Figure 3-44 and Figure 3-45.

There are two ways of reducing these interpolation errors:-

1. Addition of a stage of passive additional averaging after the initial Fourier correlation. The additional averaging should be over a timeframe of an exact multiple of ½ a cycle, so as to remove the 2nd harmonic interpolation errors, by the processes described in section 3.3.5.

2. The use of an active 2nd harmonic cancellation scheme. The measured values of amplitude and phase (V and Φ in equation (3.11) and Figure 3-42) are fed back into the measurement block. The approximate second harmonic terms due to the fundamental are then generated and subtracted before the Fourier correlation averaging stage. This removes the second harmonic term up front, so the averaging stages average predominantly DC terms, and thus the linear interpolation becomes much more accurate.

The first of these methods (passive cascaded averaging) is analysed later in section 3.9.

The 2nd harmonic cancellation scheme is based upon an algorithm described in a PLL design by Jovcic (2003). This is the only work found to date, aside from the SimPowerSystems blocks, which addresses the problems due to interpolation at low sample rates. Jovcic's scheme only needs to create a single cancelled waveform within the single-phase PLL for phase detection. Applying the 2nd harmonic cancellation to a full Fourier analysis described here requires the technique to be adapted so that both the cancellation terms determined from (3.11) to be applied. To do this, an estimate of the fundamental magnitude and phase must be fed back into the Fourier correlation. This requires additional sine and cosine calculations within each Fourier "part B" block, to be applied to the paths "A" and "B" shown in Figure 3-51. Paths "A" and "B" refer to the two correlation averages; path "A" is the signal times the sine correlation term, while path "B" is the signal times the cosine correlation term. The 2nd harmonic cancellation terms cannot be pre-calculated, so the additional burden on the CPU is significant considering that 3 analyses will be required (6 additional sine/cosine calculations) for each set of 3 phase signals such as V_a, V_b, V_c etc.[1]

The procedure for deriving the 2nd harmonic correction terms is as follows:

The signal V_{in} is assumed to be predominantly made up of the fundamental, i.e. $V_{in}=V\sin(\Phi)$ where Φ is of the form $\Phi=2\pi f$ plus an arbitrary phase offset. At any point in time, the input signal V_{in} will be approximately $V_e*\sin(\Phi_e)$ where V_e and Φ_e are the estimated magnitude and phase of the input. Care must be taken here because Φ_e must be an absolute phase relative to the positive-going zero crossing, and not simply the phase of the input relative to the correlating coefficients. The correlating Fourier coefficients are $\sin(phi\_corr)$ and $\cos(phi\_corr)$, where phi_corr is an angle rotating at the estimated frequency of the signal. During the Fourier correlation, path A evaluates as

---

[1] The 2nd harmonic cancellation in the Jovcic PLL is added in a convoluted manner by synthesising a signal made up of $\sin(50*(\Phi_e+phi\_corr))*\cos(52*(\Phi_e+phi\_corr))$ which gives components at 102*F and 2*F. The 102*F component is then filtered out using a digital low-pass filter. It is not clear why such a method is used, and it does not work at discrete sample rates as the 102*F component aliases back onto other unwanted frequencies. It is much simpler to directly synthesise the cancellation components.

$$Path\_A = V \times \sin(phi\_corr) \approx V_e \times \sin(\phi_e) \times \sin(phi\_corr)$$

$$Path\_A \approx \frac{V_e}{2}\left(\cos(\phi_e - phi\_corr) - \cos(\phi_e + phi\_corr)\right)$$

<div align="right">(3.12)</div>

which is made up of the desired DC term

$$\frac{V_e}{2}\cos(\phi_e - phi\_corr)$$

and the undesired 2nd harmonic AC term

$$-\frac{V_e}{2}\cos(\phi_e + phi\_corr)$$

For path A, the correction term is thus the negative, to cancel it out

$$+\frac{V_e}{2}\cos(\phi_e + phi\_corr)$$

<div align="right">(3.13)</div>

For path B, the analysis is similar and results in a correction term of

$$-\frac{V_e}{2}\sin(\phi_e + phi\_corr)$$

<div align="right">(3.14)</div>

The algorithm for the Fourier analysis with 2nd-harmonic cancellation is shown below in Figure 3-51, which can be compared to the un-cancelled version in Figure 3-42. There are two major additions for the 2nd harmonic cancelled block:-

1. The calculation of the cancellation terms at the 2nd harmonic, and their addition into the path averaging sections "A" and "B". This calculation is shown in Figure 3-52

2. The path averaging sections "A" and "B" are duplicated into two pairs; an un-cancelled pair and a cancelled pair. The reason for this requires significant explanation which follows below.

Discrete Fourier analysis, with a single cycle base
Uses 2nd harmonic cancellation to minimise ripple at the output

Andrew Roscoe, 2007

```
Hard math function count:-
sin/cos/tan   : 2 (Cancellation)
asin/acos     : 0
atan,atan2    : 1 (Cart to polar)
sqrt          : 1 (Cart to polar)
x^y           : 0
e^x, 10^x     : 0
ln, log10     : 0
```

To get the phase of the waveform relative to the zero crossing,
such that Mag*sin(theta) approximates the input signal,
add phi_corr to Phase

Evaluate approximate rate of change over last 1/2 cycle

This gives the phase change expected due to a ROCOF of 10 Hz/s at Fnom
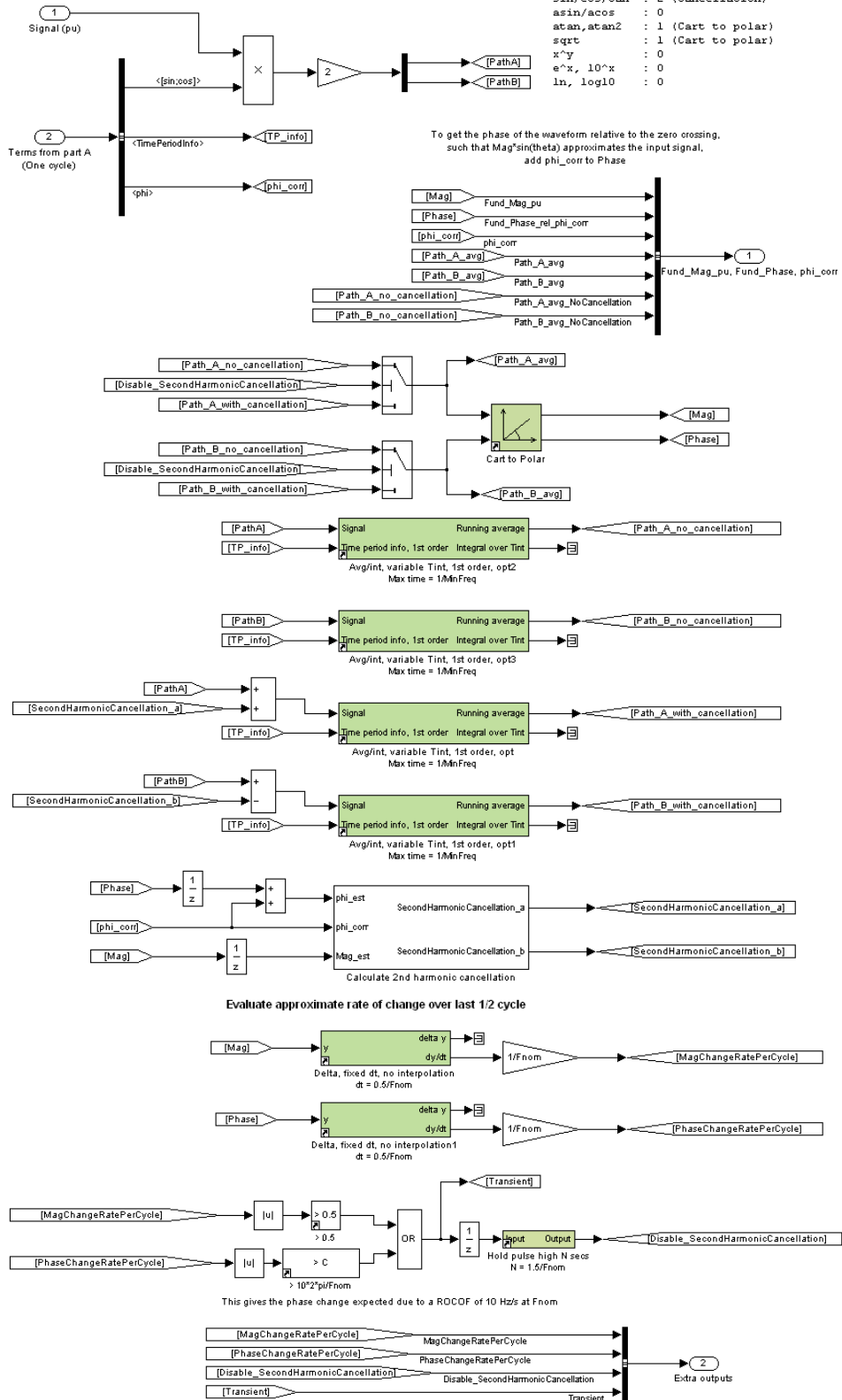
Figure 3-51 : Fourier measurement "part B" with 2nd harmonic cancellation (1st order)

125

Calculate 2nd harmonic cancellation quantities
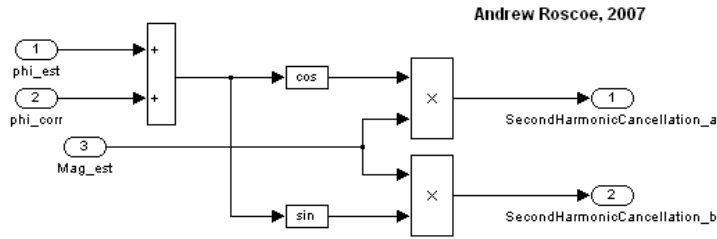
Figure 3-52 : Calculation of 2nd harmonic cancellation terms

As will be shown shortly, the ability of the 2nd harmonic cancellation algorithm to remove ripple due to interpolation of the waveform fundamental, under steady-state conditions, is extremely good. There are two weaknesses, however.

1. The 2nd harmonic cancellation technique only reduces interpolation ripple due to the fundamental signal component, and does not remove interpolation ripple which arises due to higher order harmonic components

2. The 2nd harmonic cancellation involves feeding back of the measured signal amplitude and phase into the algorithm. In this way it has an IIR (infinite impulse) response, and under transient conditions such as sudden signal amplitude change, the magnitude and phase outputs of the algorithm ring and oscillate before eventually settling.

Point 2 can be addressed for transient conditions by using the two pairs of path "A" and "B" integrators, which are shown in Figure 3-51. The idea is that when a transient is detected, the algorithm resorts to an un-cancelled mode of operation. This removes the IIR characteristics and allows the entire algorithm to settle completely within the timeframe of one averaging timeframe (1 cycle in this case). When this is complete, the cancellation mode can be re-engaged. At first glance, it would appear that this can be done with just a single pair of integrators, by feeding them with un-cancelled or cancelled signals as appropriate. However, this results in an undesirable transient in the output due to the sudden change in the inputs to the averaging filters as the second harmonic is added (or taken away). Therefore, to achieve a smooth handover, 2 pairs of integrators are required. One pair is always with-cancellation, and the other without. A mode flag determines which pair of integrators to use.

To illustrate this dynamic behaviour for a sudden increase in signal amplitude, a small Simulink model was created. This applies a 1pu signal at the worst frequency for interpolation errors (52.632Hz), then a short voltage dip to 0.25pu between 0.1 and 0.2

seconds. In this case the front-end low-pass filters, ADC sampling (but not quantisation noise) plus DC block are all modelled accurately, to show the true dynamic response at 500 Sa/s, nominally 10 Sa/cycle at 50 Hz. Several algorithms are compared here:

- The single-cycle, 1st order algorithm without cancellation from section 3.6
- A single-cycle, 1st order algorithm with cancellation always active
- The single-cycle, 1st order algorithm with the automatic cancellation decision algorithm shown in Figure 3-51

The graphs below show the same data, first in broad view and then zoomed in.
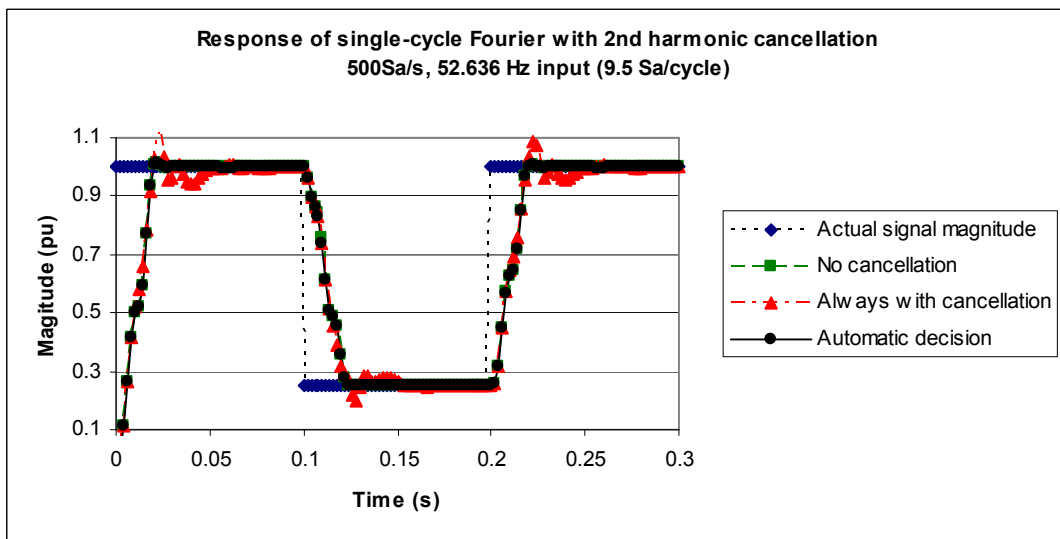
Figure 3-53 : Dynamic response of single-cycle Fourier analyses with 2nd harmonic cancellation
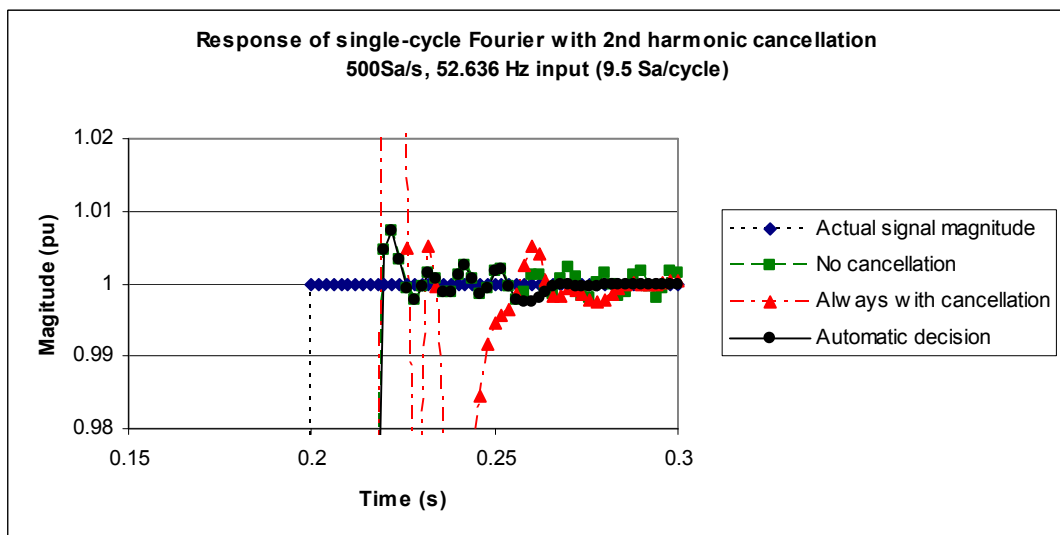
Figure 3-54 : Dynamic response of single-cycle Fourier analyses with 2nd harmonic cancellation, zoomed in.

Clearly, the "Always with cancellation" algorithm shows a poor settling characteristic, with ringing to the 0.01pu error level for 50ms (2.5 cycles) after the transient occurs. The "No cancellation" algorithm shows a much quicker settling to this level, within 20ms. These times include the low-pass filter and sampling delays, as these are modelled here. The "No cancellation" algorithm does, however, continue to exhibit a perpetual error, rippling at the 0.0016pu RMS level, at 2*52.632 Hz, due to the interpolation error. This links to the simulation results in Figure 3-44 , Figure 3-45 and Figure 3-47. The best result is the algorithm with automatic selection of cancelled and non-cancelled path averages. This shares the fast settling of the "No cancellation" algorithm, but once this is settled, it changes over to "Cancelled" operation. Because the output is already settled within about 0.001pu, the subsequent ringing of the closed-loop cancellation algorithm is not evident.

To further examine the improvement that the 2nd harmonic cancellation technique has on the errors due to integration/interpolation, the simulations of section 3.7 were modified to examine the new algorithm under the same conditions. Only the results of the 1st order algorithm is presented here.

The results are shown below, and can be compared directly to the un-cancelled algorithm performance shown in Figure 3-44 to Figure 3-50. An important point must be emphasised here. Referring back to Figure 3-53, Figure 3-54, and the text preceding these figures, the 2nd harmonic cancellation forms an IIR filter. Although the scales even in Figure 3-54 do not allow it to be observed, the output of the 2nd harmonic cancellation scheme continue to ring in a damped fashion for an "infinite" amount of time, even after implementation of the improvements for dynamic response. Thus, in the preceding section 3.7 and the following section 3.9, which analyse measurement systems without 2nd harmonic cancellation (FIR systems), the simulations only allow a fixed time period of 1½ to 2 cycles (0.03-0.04 seconds) for the measurement to settle before assessing the measurement error. In the case of the 2nd-harmonic cancellation however, the error due to a pure sinusoid input does eventually drop to 0 after a very long time. In the results presented below, the algorithm was allowed an increased settling time of 0.12 seconds (≈6 cycles). In the results which follow, for the pure sinusoidal input, the error magnitude is determined not so much by steady-state performance of the algorithm as for the FIR systems, but by the degree of settling which occurs during the 0.12 seconds allowed.

The error magnitudes shown in Figure 3-55 (and following) plots are the RMS of the instantaneous ripple error values, with the mean of the "root mean squared" evaluated over one cycle.
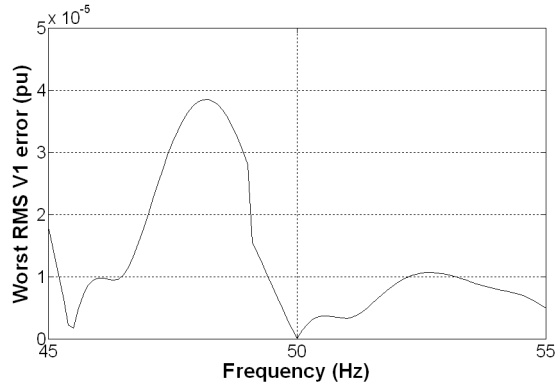
Figure 3-55 : Fourier analysis of fundamental, using 2nd harmonic cancellation. RMS errors due to integration/interpolation @ 10 Sa/cycle. 1st order method.

The RMS ripple errors on the fundamental measurement at 10 Sa/cycle are reduced from 1.6e^{-3} to 4e^{-5}, a factor of ≈40, compared to the algorithm without 2nd harmonic cancellation.
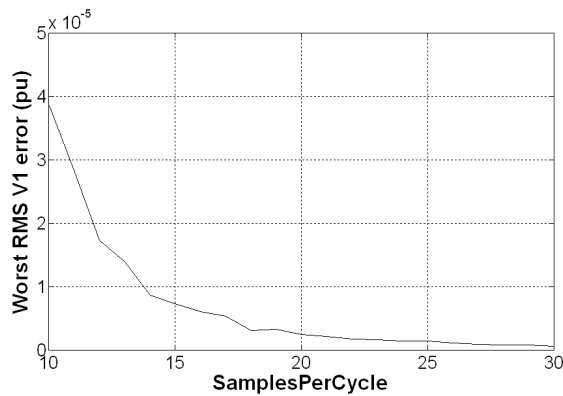
Figure 3-56 : Fourier analysis of fundamental, using 2nd harmonic cancellation. Largest RMS errors due to integration/interpolation over the 45-55Hz range. 1st order method.

Approximately the same improvement factor of ≈40 applies to higher sample rates, compared to Figure 3-45.
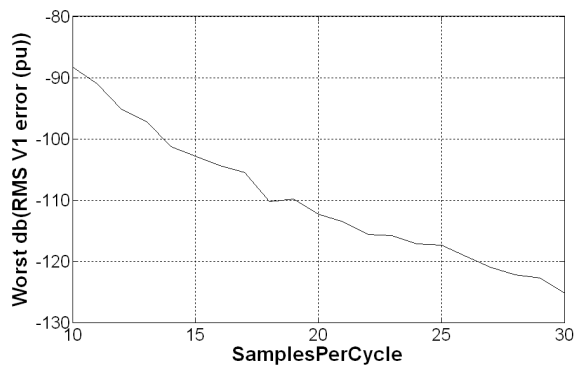
Figure 3-57 : Fourier analysis of fundamental, using 2nd harmonic cancellation. Largest dB(RMS errors) due to integration/interpolation over the 45-55Hz range. 1st order method.

By the same calculation method as shown under Figure 3-46, the error relationship follows a relationship of approximately $N^{-3.9}$
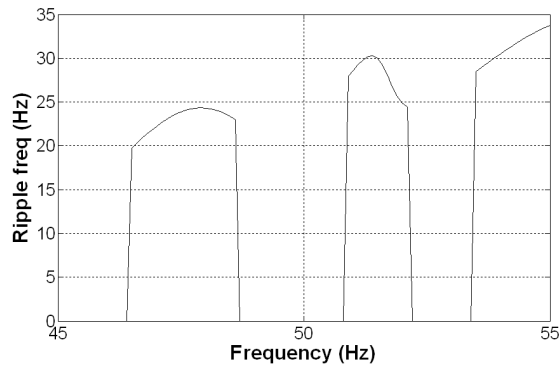
Figure 3-58 : Fourier analysis of fundamental, using 2nd harmonic cancellation. Ripple frequencies of largest errors due to integration/interpolation over the 45-55Hz range. 1st order method.

Note, however, that the frequency of the remaining error is no longer at twice the input frequency, which was the relationship shown from Figure 3-47 for the un-cancelled algorithm. This is because the ripple frequency is modified by the IIR ringing effect introduced by the 2nd harmonic cancellation. This has important (bad) consequences for the effectiveness of subsequent cascaded averaging filters.
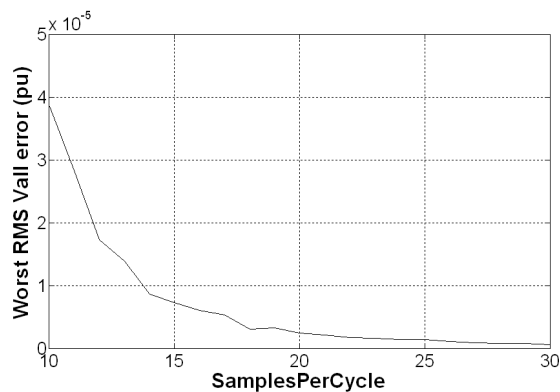
Figure 3-59 : All-harmonic analysis, using 2nd harmonic cancellation. Largest RMS errors due to integration/interpolation over the 45-55Hz range. 1st order method.

The magnitude of the ripples on the all-harmonic RMS measurement are reduced by a factor of ≈20, compared to Figure 3-49

130
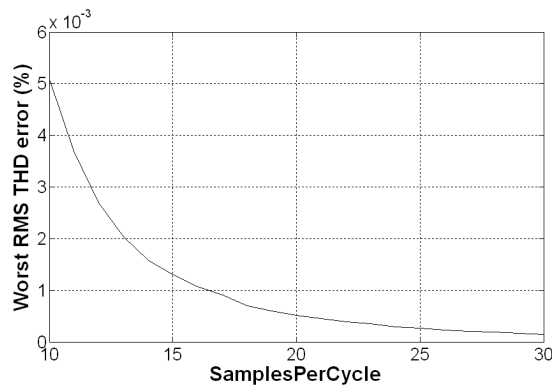
Figure 3-60 : THD analysis, using 2nd harmonic cancellation. Largest RMS errors due to integration/interpolation over the 45-55Hz range. 1st order method.

The errors on the THD measurement, for pure sinusoid inputs, are reduced by a massive factor of ≈500 compared to Figure 3-50. This is to do with a different way that THD is calculated by the 2nd harmonic cancellation algorithm. Referring back to Figure 3-43, equations (3.8) and (3.9), and Figure 3-51, the method for calculating the all-harmonic RMS value and the THD value, using 2nd harmonic cancellation, becomes:

- Calculate an estimate of the signal value at the sample time. This is done in a similar but simpler fashion to the calculation of the Path A and Path B cancellation terms. In this case, the fundamental input signal is estimated to be simply V_e*sin(Φ_e) where V_e and Φ_e are the estimated magnitude and phase of the input, fed back from the algorithm output. This value can be subtracted from the input signal to give a new value V_h. This signal V_h is the estimate of the harmonic content in real time. This is then passed through an RMS measurement block exactly the same as shown in Figure 3-43, with an integration period of 1 cycle. The output is the estimate of V_h, the RMS magnitude of all the non-fundamental components of the input signal. This leads to estimates of the all-harmonic value V_{All} and *THD*, via re-arrangement of equations (3.8) and (3.9).

It was attempted to roll this style of THD calculation back into the non-cancelled algorithms of section 3.6, but this was not effective. Further subsequent analysis shows that the extremely good result of Figure 3-60 is a particular corner case for the 2nd harmonic cancellation scheme, and the errors are so small only for the case of exactly sinusoidal inputs in steady state. Harmonic contamination, noise, or the presence of dynamic signals degrades the THD accuracy substantially.

3.8.1   Findings from this section

- 2nd harmonic cancellation can be used to significantly reduce the rippling

interpolation error at 2*f, for a 1 single cycle Fourier transform or RMS calculation output, if the input waveform has low harmonic content. The error reduction, compared to the algorithm without cancellation, is a factor of ≈40 for the fundamental measurement, and ≈20 for the all-harmonic RMS measurement, for a pure sinusoid input.

- The algorithm, being an IIR system with a feedback path, can exhibit ringing under transient conditions. A novel algorithm involving detection of transients and mode-switching between "cancelled" and "non-cancelled" operation can be used to limit this ringing.

3.9   Addition of a ½ cycle passive cascaded averaging stage

As described at the beginning of section 3.8, the ripple frequencies of the errors from a single-cycle Fourier transformation block, due to integration and interpolation, appear at twice the input frequency. So far, methods for reducing this which have been discussed are the extension to 2nd order integration/interpolation, and the addition of a 2nd harmonic cancellation scheme. In this section, a third option is described. The third option is available if the acceptable latency of the measurement inside the digital system (after low-pass filtering, sampling and pre-filtering) is 1½ cycles or more.

The idea is extremely simple but novel. Since the remaining errors due to integration/interpolation, at the output of a single-cycle un-cancelled Fourier transform block, ripple at 2*f or multiples of 2*f (if higher-order harmonics are present), then a subsequent, cascaded averaging step of ½ a cycle period will reject this ripple. This was previously described in section 3.3.5. This subsequent averaging stage also serves to further filter and bandwidth-limit the effects of noise, as described in section 3.3.4.

The algorithm described here and shown below can be named in an abbreviated way as "1(NC)+0.5, 1st order", which means that a 1-cycle (Non-Cancelled – without 2nd harmonic cancellation) base measurement (Fourier transformation and/or RMS/THD calculation) is followed by an additional ½ cycle averaging, where all the averaging blocks are implemented using 1st order integration/interpolation. This algorithm builds directly on that shown in Figure 3-42, which is included as a library block within the "1(NC)+0.5, 1st order" block.
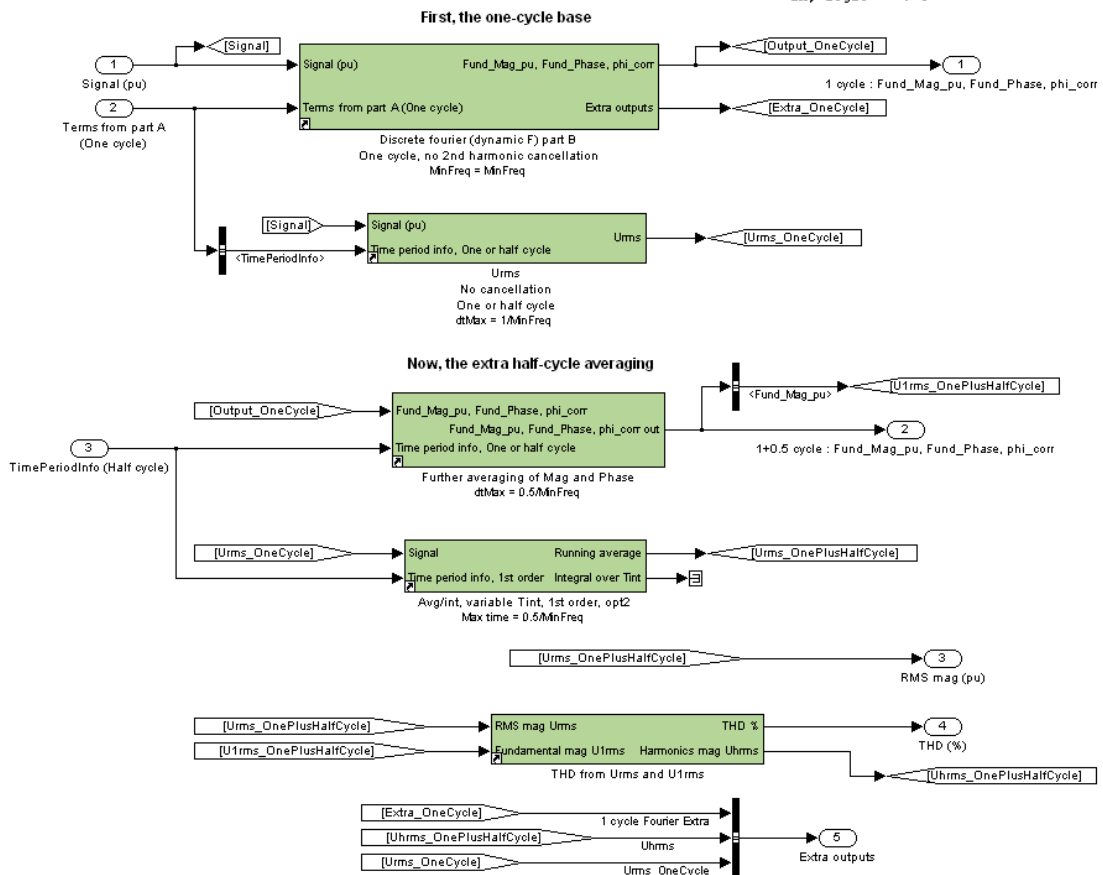
Figure 3-61 : 1(NC)+0.5 measurement algorithm detail, 1st order

Figure 3-61 is relatively self-explanatory, and most of the lower level blocks have either been previously described or contain only basic THD calculations. There are 3 points of note:-

- The algorithm outputs measurements of the fundamental magnitude/phase & RMS on both a 1-cycle and a 1+0.5 cycle basis. These can be used as appropriate for subsequent functions which may prioritise measurement speed (use the 1 cycle output) or measurement ripple minimisation (use the 1+0.5 cycle output). The THD calculation is only carried out on the 1+0.5 cycle basis, since THD is not required for fast relaying actions.

- The algorithm above applies the extra ½-cycle averaging to the magnitude and phase outputs from the first 1-cycle Fourier correlation. A very marginal reduction in ripple at the 1+0.5 cycle output can be achieved by instead averaging the "Path A & B" averages from the Fourier transform block (see Figure 3-42, and also

Figure 3-69). In some circumstances this is not the best option, however, since it requires an additional Cartesian-to-polar transformation in the ½ cycle averaging, which is reasonably CPU-intensive as it uses both an "atan2" and a "sqrt" function. It should be mentioned, however, that this method must be used for extra ½ cycle averaging if the initial Fourier block implements 2nd harmonic cancellation as described in section 3.8. This option is described in section 3.10. Also, and of most relevance, use of this "path averaging" technique can actually reduce the overall CPU loading, despite the extra up-front calculation at this stage. The reason for this is that common operations subsequent to the Fourier calculations are sequence analysis (+ve, -ve, zero) for 3-phase voltage waveforms, and also power calculations (P, Q & S), from 3-phase voltage and current sets. It turns out that doing the path averaging means that many sine/cosine evaluations can be saved later on, by re-using the results of the path average directly, in combination with the magnitude and phase of the final result. The sine and cosine of the voltage and current phase angles can be deduced directly from the path A and path B averages and the hypotenuse (voltage magnitude), by the use of simple division operations.

- Inside the "Further Averaging of Mag and Phase" block, of Figure 3-61, the averaging of phase requires careful implementation to avoid problems with phase wrapping at the $-\pi$ and $+\pi$ boundaries (see section 3.2.4).

As in the previous section, the simulations of section 3.7 were again modified to examine the new algorithm under the same conditions. Results from both the 1st and 2nd order algorithms are presented below. These can be compared directly to the un-cancelled performance in Figure 3-44 to Figure 3-50 and the 2nd-harmonic cancellation performance in Figure 3-55 to Figure 3-60.

The error magnitudes shown in Figure 3-62 (and following) plots are the RMS of the instantaneous ripple error values, with the mean of the "root mean squared" evaluated over one cycle.
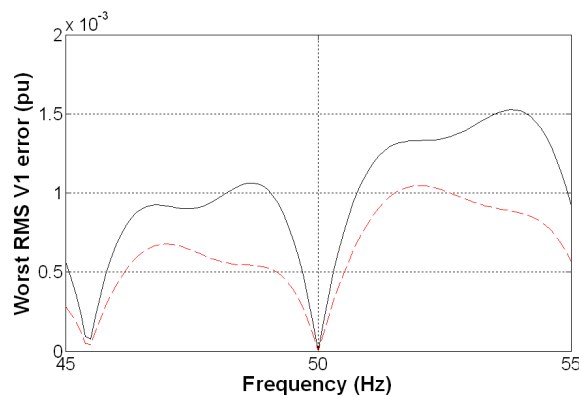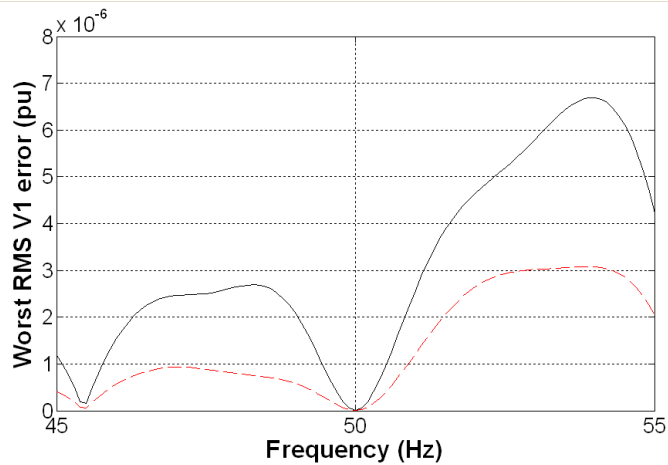
Figure 3-62 : Fourier analysis of fundamental. RMS errors due to integration & interpolation @ 10 Sa/cycle. 1st order (solid) and 2nd order (red dashes) methods with additional half-cycle averaging
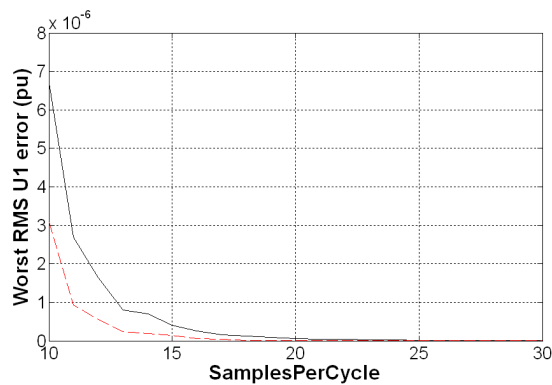
Figure 3-63 : Fourier analysis of fundamental. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods with additional half-cycle averaging
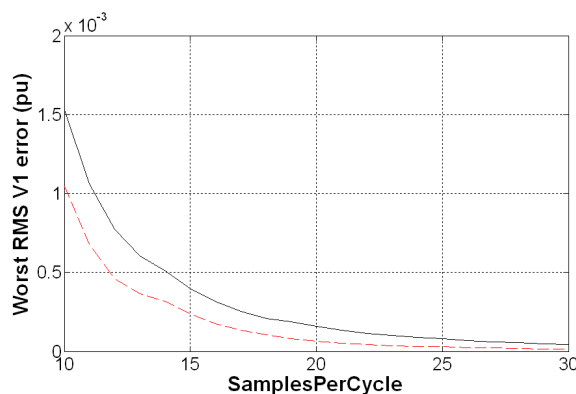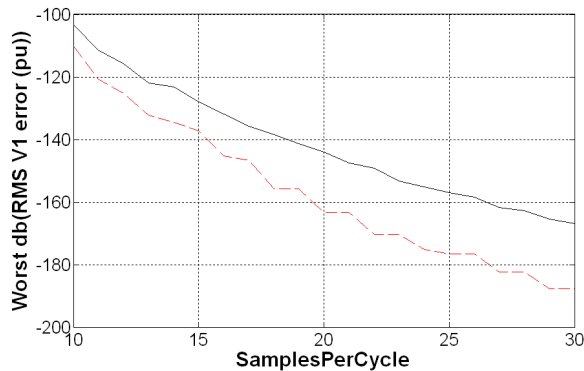

Figure 3-64 : Fourier analysis of fundamental. Largest dB(RMS errors) due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods with additional half-cycle averaging

Following the calculations under Figure 3-46 & Figure 3-57, the error magnitude follows a relationship of approximately $N^{-6.7}$ for the 1st order method, and N^{-8} for the 2nd order method.
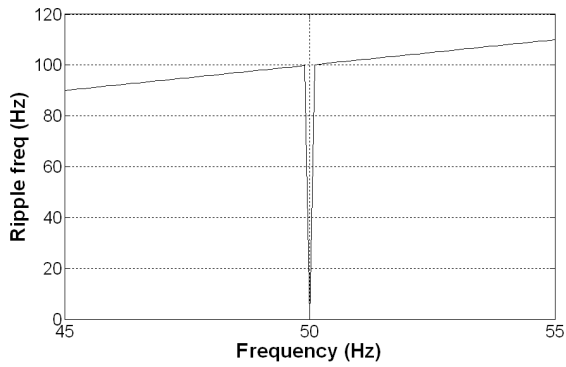
Figure 3-65 : Fourier analysis of fundamental. Ripple frequencies of largest errors due to integration & interpolation over the 45-55Hz range. 1st order method with additional half-cycle averaging
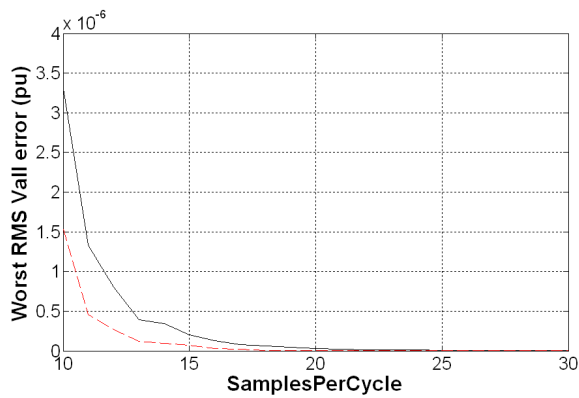
Figure 3-66 : All-harmonic analysis. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods with additional half-cycle averaging
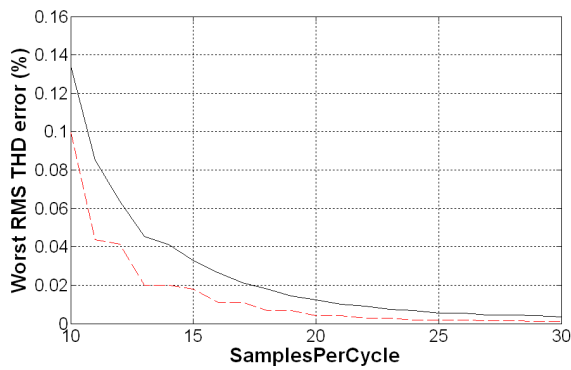
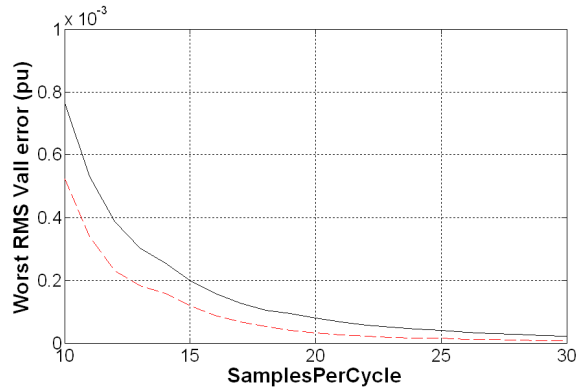**Figure 3-67 : THD analysis. Largest RMS errors due to integration & interpolation over the 45-55Hz range. 1st order (solid) and 2nd order (red dashes) methods with additional half-cycle averaging**

136

3.9.1   Findings for this section

- The integration and interpolation errors for the Fourier fundamental and all-harmonic RMS measurements are practically eliminated, to less than <0.00001pu, for all values of samples per cycle ≥10 Sa/cycle, and for both 1st and 2nd order methods. The reduction in ripple due to integration/interpolation, compared to the standard 1-cycle un-cancelled algorithm, is a factor of ≈200

- If a measurement latency within the digital system of 1½ cycle can be tolerated for precision measurements, then the interpolation ripple error performance of the "1(NC)+0.5, 1st order" algorithm surpasses the 1-cycle 2nd harmonic cancellation algorithm performance by a factor of ≈5, while requiring substantially less expensive CPU operations due to the lack of need for 2nd harmonic waveform generation.

- With these algorithms, there is no need (or benefit) in synchronising the samples with the zero crossings by using expensive variable sample rate hardware, locked to the fundamental. Indeed, a non-integer number may be chosen for the nominal value of samples per cycle, without any significant detriment to the measurement quality due to integration and interpolation errors.

- The THD measurements are reduced to the <0.2% level, which is an acceptable error level.

- It has been shown that the "1(NC)+0.5" system, both 1st and 2nd order versions, produces excellent results with a well-defined FIR response which settles fully within 1½ cycles The 1½ cycle output errors due to integration/interpolation errors on the measurements of fundamental and all-harmonic RMS amplitude for pure sinusoid inputs are at levels less then -100dB(pu), even for sample rates as low as 500 Sa/s, i.e. 10 Sa/cycle @ 50Hz.

3.10   The combination of 2nd harmonic cancellation with extra ½ cycle averaging

To try and create an optimal integrated measurement algorithm, which provides both quick measurement on the 1-cycle timeframe, plus the option of slower but more accurate (less noise & ripple) measurements, the obvious temptation is to try and combine the benefits of the 1-cycle 2nd harmonic cancellation algorithm with the 1(NC)+0.5 cycle algorithm (1 cycle base measurement, Non-Cancelled, with additional ½ cycle averaging) from section 3.9. An additional idea explored was to use the output of the additionally-averaged outputs to feed back as the amplitude and phase estimates used to form the 2nd harmonic cancellation within the base Fourier stage. The idea here is that the further averaged results would have less noise and ripple, and thus the initial 2nd harmonic

cancellation stage would perform even better.

It turns out that these ideas do not provide reductions in noise or ripple. Substantial simulations were performed before this was fully understood. These do not need to be presented here in detail, since the reasons for the disappointing performance can be explained (with some hindsight) by referring back to Figure 3-47 and Figure 3-58.

The outputs of the non-cancelled 1-cycle base measurement stages contain integration/interpolation ripple at 2 times the fundamental frequency, as shown in Figure 3-47. Thus, the "1(NC)+0.5" system is able to reduce the integration/interpolation ripple to very low levels by applying an extra averaging stage of ½ cycle period duration (see Figure 3-63). This is not the case when the base measurement stage uses active 2nd-harmonic cancellation. The ripple due to integration/interpolation errors from the initial 1-cycle base stage is at a variety of frequencies as shown in Figure 3-58. This is due to the IIR characteristics of the 2nd-harmonic cancellation feedback system. Thus, an additional ½-cycle averaging does not reduce the ripple so effectively. So, although the ripple from the 1-cycle base stage with 2nd harmonic cancellation is less than the equivalent non-cancelled 1-cycle base stage without cancellation, the "1(C)+0.5" system does not give an overall better performance at its 1½ cycle output port than the "1(NC)+0.5" system. Note that in the case of a pure sinusoid input at fixed frequency only, a "1(C)+0.5" system can give the best results. However, as soon as any harmonic content, noise or dynamically changing conditions arise, the "1(NC)+0.5" system outperforms the "1(C)+0.5" system. The performance of the "1(C)" and "1(C)+0.5" algorithms under such conditions is compared to other algorithms in section 3.13.

Feeding back the additionally-averaged magnitude and phase outputs into the 2nd-harmonic cancellation was also found not to be of any extra benefit over the 1-cycle feedback process shown in Figure 3-51, to counteract the effects of either harmonic contamination or noise on the input signal. The additional delay in the feedback loop also degraded the performance of the algorithm under dynamic conditions from the response shown in Figure 3-53 & Figure 3-54, which is highly undesirable.

Therefore, if it is desired to obtain a 1-cycle output with 2nd-harmonic cancellation, plus also a more accurate result in 1½ cycles, the best system is a parallel pair of systems, with a 1(C) system and a totally separate 1(NC)+0.5 path. Such an algorithm is shown below, of the 1st order variety. The RMS and THD calculations are only output on the 1½ cycle timeframe using the non-cancelled path, since this removes the need for the alternative

RMS/THD derivation which is described under Figure 3-60, and thus saves an expensive sine operation, plus several delay buffer blocks.
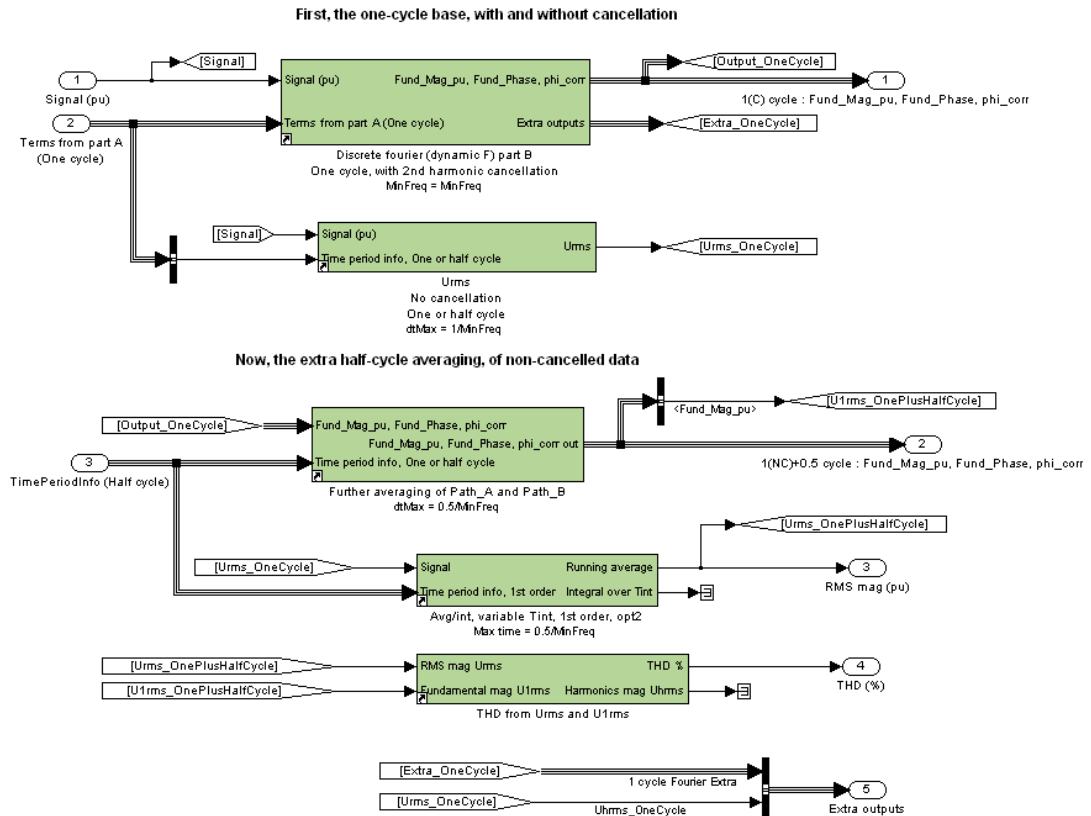
Figure 3-68 : Combined 1(C) and 1(NC)+0.5 measurement algorithm

All the blocks within this algorithm have previously been described, except for the "Further averaging of Path_A and Path_B" block (although it was referred to in the findings to section 3.9). This is shown below. Note that this requires an additional atan2 and sqrt function, and is thus reasonably CPU intensive. Over and above the un-cancelled "1(NC)+0.5" system, which requires 4 "hard" math operations[1], the "1(C)" in parallel with the "1(NC)+0.5" system requires 8 "hard" maths operations. 2 of these are the sine/cosine for the 2nd-harmonic cancellation, and 2 of these are required for additional path

---

[1] See section 3.13.1, Table 3-3 for the list of "hard" maths functions

139

atan2/sqrt functions required for the Cartesian to polar transformation inside the extra path averaging.
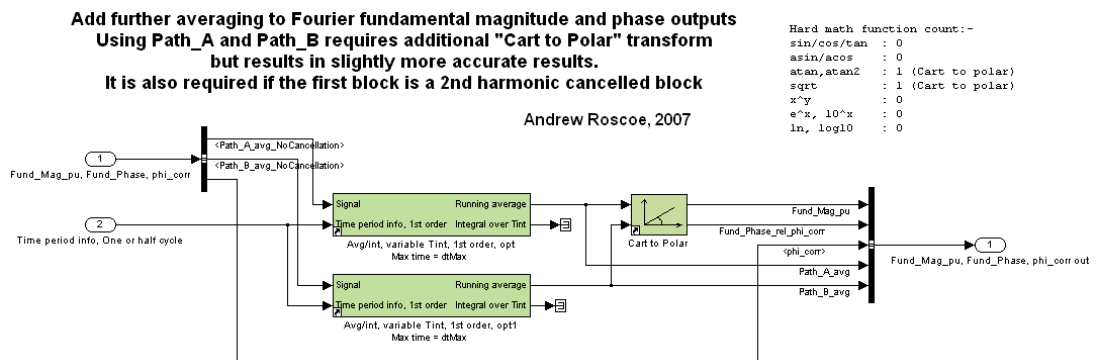
Figure 3-69 : Path averaging for the combined 1(C) and 1(NC)+0.5 measurement algorithm

3.10.1 Findings from this section

- The ripple from the 2nd-harmonic cancelled base measurement blocks is not always at 2*Fundamental or harmonics of this frequency. This is due to the IIR response of the algorithm.

- A subsequent ½ cycle averaging stage does not, therefore, have anywhere near as much additional benefit as it does for the un-cancelled 1-cycle base measurements.

- Where the most accurate 1-cycle base measurements are required, the 2nd harmonic cancelled algorithms are the best (requiring extra CPU effort to carry out this task).

- Where measurement latency in the digital system of 1½ cycles or more can be tolerated, then either the non-cancelled path of the algorithm should be used for further averaging, or a non-cancelled base stage should be used.

- If minimising ripple is not of the highest priority from the initial 1-cycle base measurement, then algorithm execution time, complexity, and robustness during dynamic events can be improved by not using the 2nd-harmonic cancelled base stage, and using the simpler un-cancelled base stage of section 3.6.

3.11   Base measurement stages of ½ cycle duration

The mathematical expressions for Fourier and RMS measurements allow the measurements to be made over timeframes other than 1 cycle. Normally, however, it would be expected that the timeframe of the measurements would be >=1 cycle, and often the measurement time cannot be set to exact multiples of the fundamental cycle period as is being done in this thesis. In this case, Fourier windowing techniques such as Hanning windows are used

to minimise the ripple magnitude due to the effect of non-integer numbers of cycles appearing inside the analysis window.

The analysis of power systems waveforms is rather a specialised field because the desire is to measure the size/phase/frequency of a nominally sinusoidal waveform with latencies of the order of <1 to 10 cycles. For protective relaying of overcurrent, measurement latencies of <1 cycle (or as small as reasonably possible) are desirable. To do this, techniques such as the 2 and 3-sample algorithms from Johns (1995) have been developed. These can give an estimation of signal amplitude in <<1 cycle for sinusoidal signals. These kinds of techniques can be used on HV & EHV transmission lines where the THD is low, and fast tripping times are very important since the protection is likely to be of the "unit" variety, designed primarily to protect the line from melting/sagging which it may do very quickly under fault conditions due to the low per-unit impedance.

Under the influence of harmonic contamination and flicker levels such as those described in section 2.7, however, the outputs such sub-cycle algorithms become corrupt. Within a microgrid context, the scenario and priorities are different. THD on the voltage & current waveforms is much higher, but longer tripping times can be tolerated since the protection will be part of a graded protection scheme, the per-unit impedances are higher, and the distribution/switching/breaking/protection equipment will be designed to carry rated fault current for certain times without damage. This means that a measurement latency of << 1 cycle (much less than one cycle) is not required for protection purposes within a microgrid. However, a <1 cycle measurement latency (somewhat less than a cycle digital response time) may be desirable, especially when the minimum achievable latency is limited by the filtering and ADC sampling latencies shown in Figure 3-30 & Figure 3-31.

An intermediate solution, between a <<1 cycle and a 1-cycle measurement, is a Fourier or RMS calculation over exactly ½ cycle (Johns, 1995). To do this, any of the algorithms from section 3.6 or section 3.8, 1st or 2nd order, can be adapted to operate with on a ½-cycle base measurement basis. These algorithms have been coded as Simulink blocks and analysed in detail. The code does not need to be presented or described here due to its total similarity to those algorithms already described. Only the time periods for all the base stage averaging processes are different, being halved.

The performance of some of these blocks is presented together with other blocks in section 3.13. Here, it is useful to describe their properties in words. Figure 3-70 to Figure 3-77 show plots which corroborate these statements.

3.11.1 Properties of the ½ cycle base measurement stage

- The integration/interpolation errors are larger than for the 1-cycle equivalent base stages. This is not because the interpolation errors are larger, but because the timeframe is half as long which leads to an average calculation in which the interpolation error is not "spread" over as long a timeframe. The worst frequencies for integration/interpolation errors occur at

$$Fworst = \frac{F_{nom} \dfrac{N}{2}}{\left(\dfrac{N}{2} \pm \dfrac{m}{2} \right)}$$

(3.15)

where m is any sensible odd integer to give a positive frequency. This can be compared directly to equation (3.10). The worst frequencies are roughly twice as far apart as for the 1-cycle base measurements.

- Theoretically, odd harmonics present on the input signal will be rejected by a ½-cycle analysis. However, odd harmonics do increase the interpolation error (as they do for the 1-cycle base measurements).

- Even harmonics present on the input signal will not be rejected by a ½-cycle analysis. Thus, any input signal containing even harmonics will produce a ripple at the output of the ½-cycle block. This ripple consists of the theoretical ripple output due to the harmonic, plus the additional integration/interpolation ripple.

- DC offsets on the input signal are a special case of an even harmonic. They are effectively the 0th harmonic, and a DC level of *x* produces a theoretical ripple magnitude equivalent to that of any non-DC even harmonic at a peak amplitude of *2x*. Thus, even small DC offsets can lead to large ripples at the output of the ½-cycle base stages. Fortunately, this effect can be ignored for the practical applications described in this thesis, as DC offsets can removed by the DC blocking filter designed in section 3.4.4.2. The exception is during a hard fault when the current or voltage waveforms may contain genuine DC offsets for a few cycles. During this time, the DC blocking filter will allow the DC through, so a user of such ½-cycle blocks should allow for this.

- It is possible to use a 2nd-harmonic cancelled version of the ½-cycle base measurement block. This, however, is susceptible to relatively large ripple/ringing when even harmonics (or stray DC) are applied (see Figure 3-70). Its performance under these conditions is worse than the non-cancelled version. Thus, this is not a good candidate algorithm to use within the microgrid context.

- Half-cycle base measurement stages can also be used as the core of fast-responding PLLs. These are of course noisier than 1-cycle PLLs, and susceptible to ripple due to the presence of even harmonics.

3.12   "0.5(NC)+1" systems

In the previous section, the ½-cycle base measurement was introduced. This offers a faster measurement than the 1-cycle base measurement for protective over-current relaying purposes. The 2nd-harmonic cancelled version of the ½-cycle base block has not been found to be a sensible algorithm to use within the microgrid context, due to signal contamination by even harmonics. The un-cancelled ½-cycle base measurement, however, can be used as the base for accurate measurements. The idea is that the simple, un-cancelled ½-cycle base measurement is applied first. This will output more ripple than the equivalent 1-cycle base measurement, both due to integration/interpolation errors of the fundamental and harmonics, plus much larger ripples due to even harmonic and DC contamination. The signal output, even with this ripple added, will still be usable for fast-acting protective over-current relays with trip settings high enough above 1pu to avoid spurious tripping due to the ripple.

The ripple which is output from the ½-cycle base measurement blocks turns out to be at frequencies of multiples of $1*f$. Now, recalling that the 1-cycle base measurements contain ripple at $n*2*F$, which can be almost entirely removed by further averaging over a time period $1/(2*F)$, it can be seen that the ripple from the ½-cycle base measurements can be almost totally removed by further averaging over a time period of exactly $1/f$. Thus, where a "1(NC)+0.5" system which outputs virtually no ripple due to sampling effects was created in section 3.9, a new algorithm which can be called "0.5(NC)+1" appears to be an equally valid method to achieve robust measurements at low sample rates, with virtually zero integration/interpolation errors. The classification "0.5(NC)+1" here means "½-cycle base Fourier measurement (no 2nd-harmonic cancellation) followed by 1-cycle averaging". This algorithm set has been coded in Simulink, and the results are compared with other methods in the next section.

3.12.1   Findings from this section
- The "0.5(NC)+1" measurement system is a viable measurement system, offering almost identical performance to the "1(NC)+0.5" system in the presence of harmonics and noise.

3.13 Summary of viable measurement systems and their comparisons under conditions of harmonics and noise

Thus far in section 3, several competing measurement systems have been proposed for the measurement of Fourier amplitude, phase, RMS & THD. Most emphasis is placed upon the Fourier measurements, but the algorithmic processes and relative errors have also been examined for RMS and THD measurements. This section briefly reviews the proposed systems, and compares their performance under a suite of test conditions. The viable methods proposed thus far are:-

Classification	1(NC)	1(C)	1(NC)+0.5	0.5(NC)	0.5(NC)+1
Base measurement (Fourier or RMS/THD) cycles	1	1	1	½	½
Active 2nd harmonic cancellation within the base measurement ?	No	Yes	No	No	No
Further averaging (of non-cancelled path averages for base stages with 2nd harmonic cancellation)	-	-	½ cycle	-	1 cycle
For	Robust, simple	Lowest possible ripple for sinusoidal input, with a 1-cycle measurement time. 1(NC) outputs also available.	Very low ripple for all un-aliased harmonics on input	Fast response for over-current relaying	Very low ripple for all un-aliased harmonics on input
Against		Extra CPU overhead, complexity. Cancellation not as effective when harmonics present.		Large ripples on output for even harmonics or DC on input	
Worst RMS Ripple (pu) on Fourier amplitude measurements due to a pure sinusoid input, for 1st order algorithms. [2nd order algorithms up to a factor of 2 smaller]	0.0016	0.00004	0.000007	0.0035	0.000008

Table 3-1 : Viable measurement architectures

Plus, the 6th final option is available of using a 1(C) base measurement stage, but following this through to obtain the "1(NC)+0.5" solution, since the "1(NC)" algorithm is a subset of the "1(C)" algorithm, and the "1(NC)" outputs are therefore available from the "1(C)" algorithm for further averaging.

These 6 viable options, plus the un-recommended "0.5(C)" option (for comparison only), are now subjected to further rounds of testing. This testing involves addition of Gaussian noise and harmonic contamination, and examination of the resulting worst RMS ripple errors of the algorithms for any input frequency in a range wide enough to catch the worst expected integration/interpolation errors. Two sample rates are examined: 500 Sa/s (10 Sa/cycle @ 50 Hz), and 1000 Sa/s (20 Sa/cycle @ 50Hz). The examined input frequency ranges are 44 to 56 Hz for the 500 Sa/s case, and 47 to 53Hz for the 1000 Sa/s case. This allows for full coverage of the worst frequencies for all systems with both the 1-cycle and ½-cycle base measurements, by equations (3.10) & (3.15).

The Gaussian noise simulates ADC quantisation at the 0.000282pu RMS level, or the worst case 0.005pu RMS instrumentation noise level anticipated for a voltage measurement channel (see section 2.9). The harmonics added in this round of testing are un-aliased. For the 500 Sa/s case, the 2nd and 3rd harmonics are considered. For the 1000 Sa/s case, the 2nd, 3rd and 5th harmonics are considered. The aliased harmonics have a different effect on the performance of all the blocks, which is examined later in section 4.2. In this set of simulations, the anti--alias filter response is not modelled so as to focus purely on the algorithm integration/interpolation errors. The DC block algorithm, however, is included into the simulation, to validate its performance (see section 3.4.4.2). A 0.02pu DC bias is applied to all measurement inputs.

The harmonic levels for the contaminating harmonics are chosen to be 2 times the BS EN 50160 specification for long-term average values:-

Harmonic	Harmonic frequency for 50Hz nominal fundamental	BS EN 50160 specification	Applied level in this analysis2
2nd	100 Hz	2%	4%
3rd	150 Hz	5%	10%
5th	250 Hz (therefore is aliased for 500 Sa/s systems when nominal frequency > 50 Hz)	6%	12%

Table 3-2 : Harmonic levels applied for analysis of viable blocks

The analysis is carried out by a Simulink model containing the appropriate signal generation and analysis library blocks. The analysis and spreadsheet includes results for Fourier amplitude/phase errors, RMS errors and THD errors. Only the Fourier amplitude error measurements are shown here for brevity. The relative sizes of the other types of

errors are broadly proportionate to the Fourier amplitude errors, so the decision process for the selection of the best blocks can reasonably be shown using only the Fourier errors.
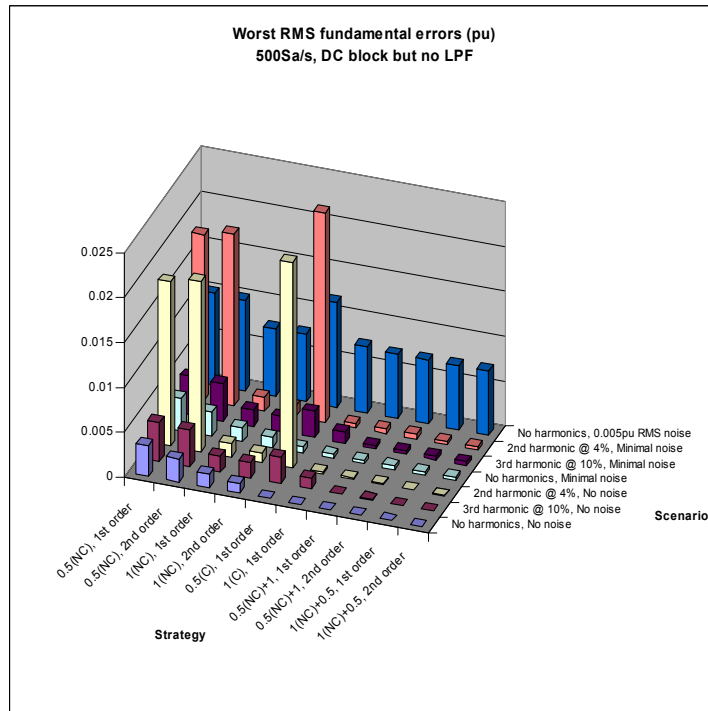
Figure 3-70 : RMS errors on the Fourier measurement of fundamental amplitude, 500Sa/s, all viable options and scenarios
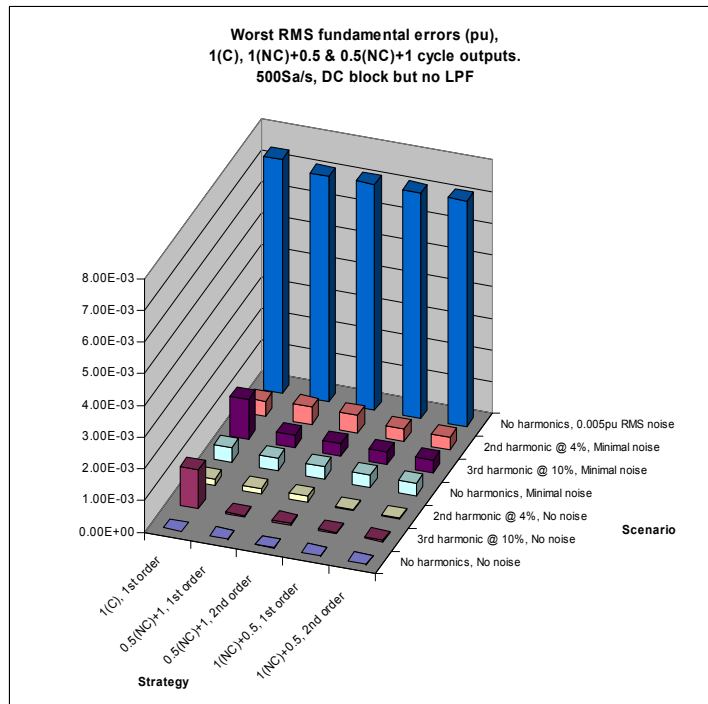
Figure 3-71 : RMS errors on the Fourier measurement of fundamental amplitude, 500Sa/s, lowest rippling options, all scenarios
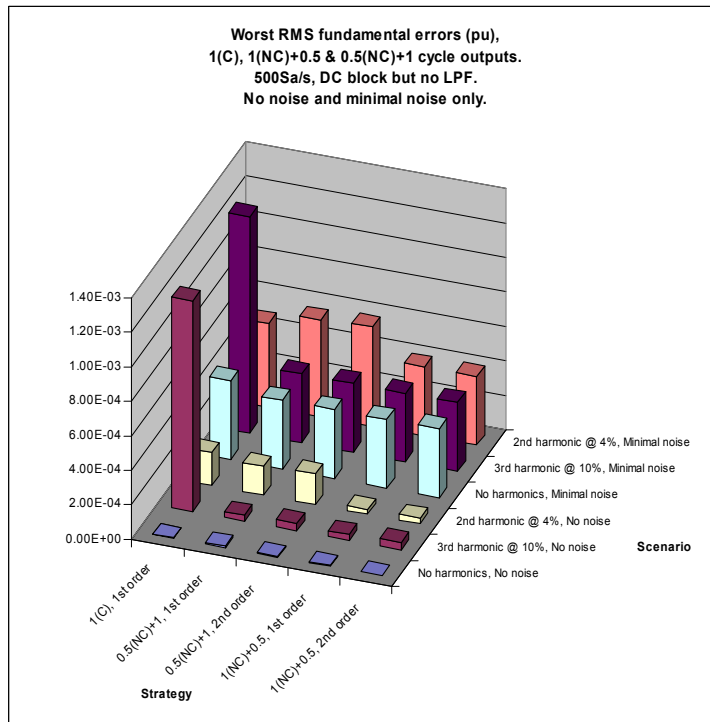
Figure 3-72 : RMS errors on the Fourier measurement of fundamental amplitude, 500Sa/s, lowest rippling options, low noise scenarios
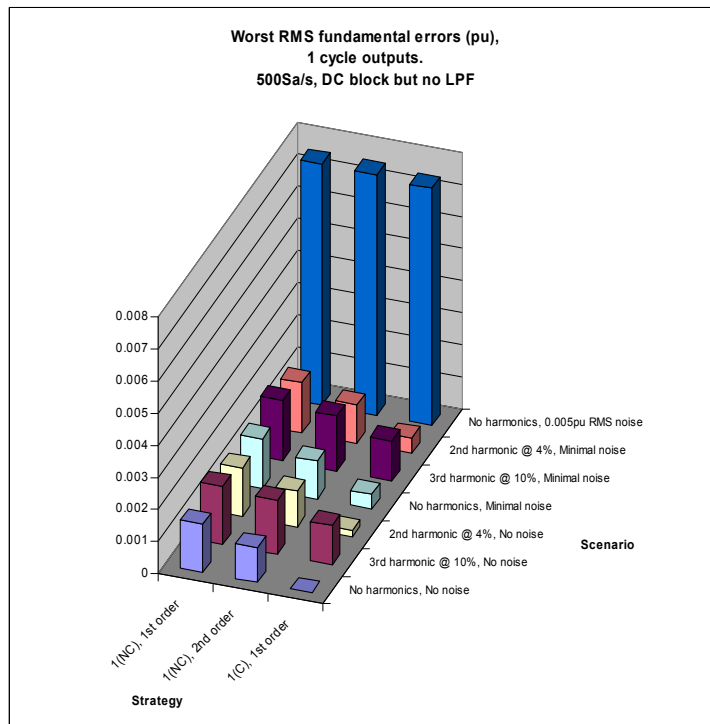
Figure 3-73 : RMS errors on the Fourier measurement of fundamental amplitude, 500Sa/s, 1-cycle latency measurements, all scenarios
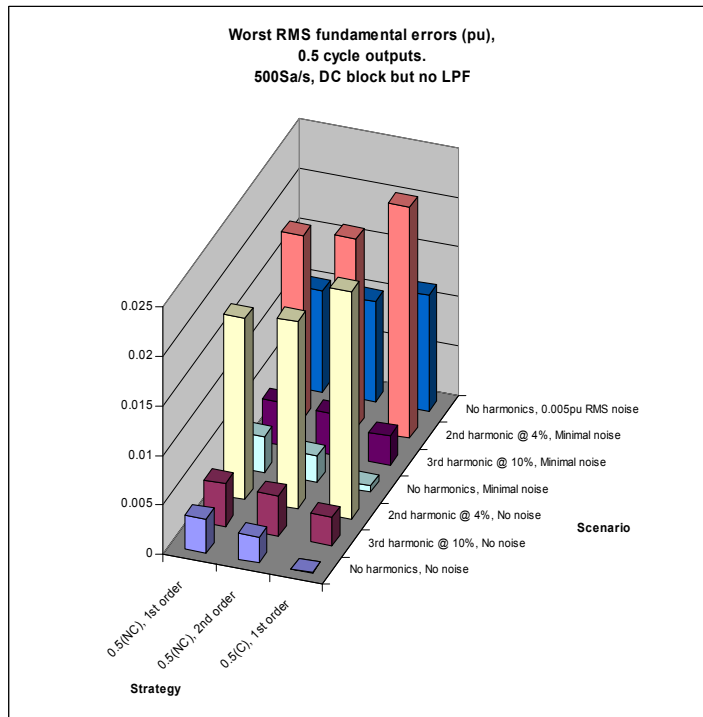
Figure 3-74 : RMS errors on the Fourier measurement of fundamental amplitude, 500Sa/s, ½-cycle latency measurements, all scenarios
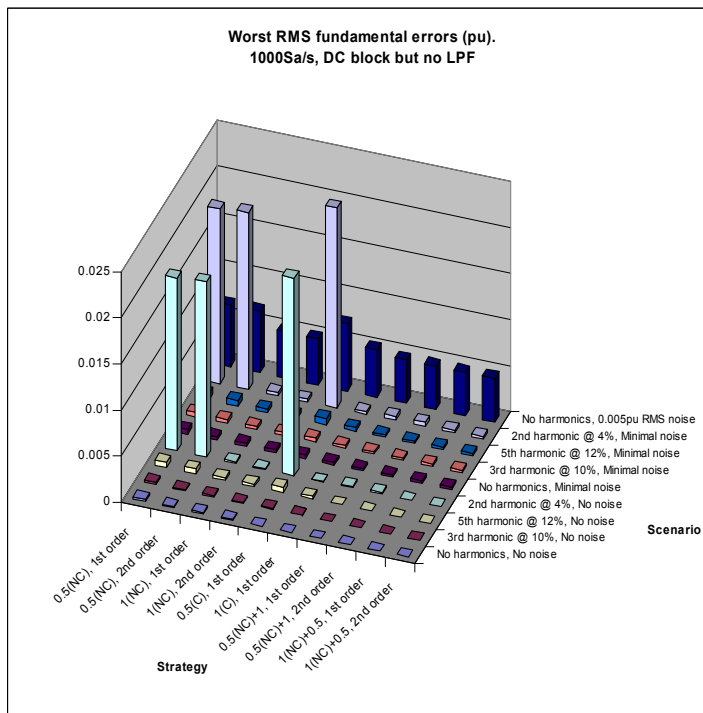
Figure 3-75 : RMS errors on the Fourier measurement of fundamental amplitude, 1000Sa/s, all viable options and scenarios
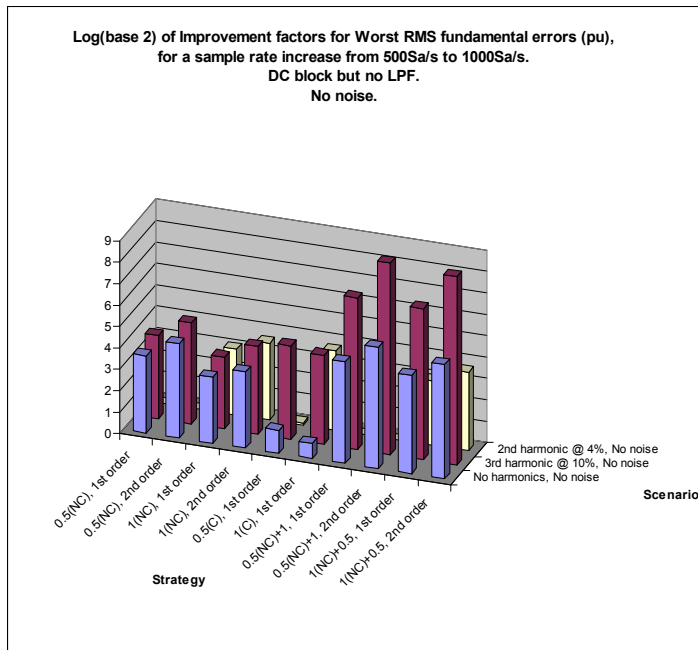
Figure 3-76 : Log(base 2) of Improvement factors for Worst RMS fundamental errors (pu), for a sample rate increase from 500Sa/s to 1000Sa/s. DC block but no LPF. No noise.
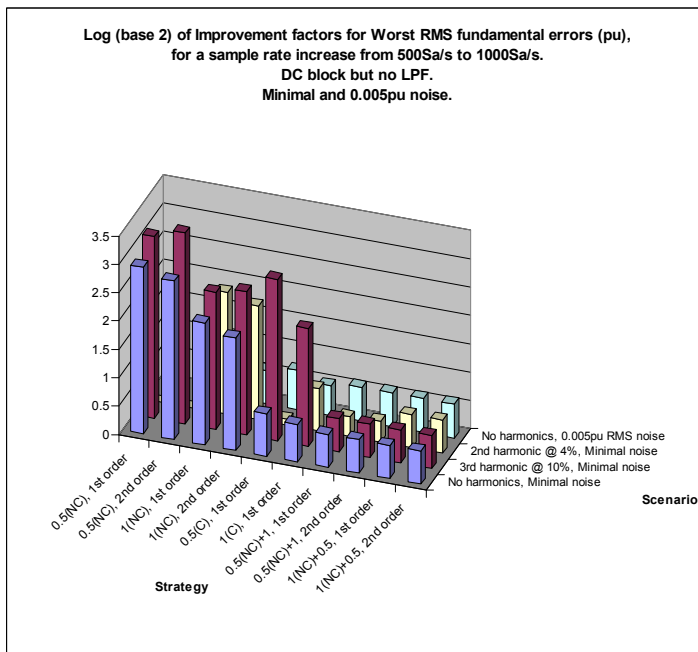
Figure 3-77 : Log (base 2) of Improvement factors for Worst RMS fundamental errors (pu), for a sample rate increase from 500Sa/s to 1000Sa/s. DC block but no LPF. Minimal and 0.005pu noise.

The data shown in Figure 3-70 to Figure 3-77, together with that of Figure 3-46, Figure 3-57, & Figure 3-64, enables some deductions to be made about the relative merits of the different measurement architectures:-

149

- The ½-cycle block with cancellation "0.5(C)" shows worse performance than the standard ½-cycle blocks under the influence of 4% 2nd harmonic, so is not worthwhile. For the fastest output, the "0.5(NC)" block is the best.

- The 1-cycle block with cancellation "1(C)" is worthwhile, particularly when the level of harmonic contamination is low. Importantly, its performance never appears to be worse than the standard 1-cycle block without cancellation "1(NC)".

- When harmonic contamination is at the higher levels of these scenarios, the "1(NC)" block performance is as good, or almost as good as the "1(C)" blocks.

- When noise rises to the 0.005pu level, it becomes the dominant factor (apart from the ½-cycle measurements affected by even harmonics), and all advantages of either the 2nd-harmonic cancellation or the 2nd order integration/interpolation is removed. This is true for both the 10 Sa/cycle (500 Sa/s) and 20 Sa/cycle (1000 Sa/s) cases.

- As sample rate is increased from 10 Sa/cycle (500 Sa/s) to 20 Sa/cycle (1000 Sa/s), the magnitude of the integration/interpolation errors from the 1-cycle, 1st order, non-cancelled block "1(NC), 1st order" drop by a factor of ≈8, i.e. as a function of ≈N^{-3}. (This corroborates with Figure 3-46, which approximately follows an N^{-3} curve). The relationships for the other blocks follow approximately: "1(NC), 2nd order", N^{-4}, "1(C), 1st order", N^{-4}, "1(NC)+0.5, 1st order", N^{-6}, and "1(NC)+0.5, 2nd order", N^{-8}. The "0.5(NC)+1" blocks behave the same way as the "1(NC)+0.5" blocks. The caveat here is that when the number of samples is very low, the errors change faster than these relationships would imply. This is shown by slight non-linearities in Figure 3-46, Figure 3-57, & Figure 3-64. Also, this effect manifests itself in poorer rejection of some un-aliased harmonics, when the number of samples in each cycle of the harmonic drops below about 10. This effect shows up well on Figure 3-76, where the rejection of the 3rd harmonic improves by significantly more than the expected amounts of 2^6 and 2^8, by 2^7 and 2^9 in fact, for the "1(NC)+0.5" and "0.5(NC)+1" blocks, 1st and 2nd order respectively, when sample rate is increased by a factor of 2.

- As sample rate is increased from 10 Sa/cycle (500 Sa/s) to 20 Sa/cycle (1000 Sa/s), the errors due to noise drop by a factor of only ≈$\sqrt{2}$, as the reduction is simply due to the number of samples in the averaging, and the application of the "random walk" effect.

- Therefore, as the number of sample-per-cycle N increases, the relative importance of the integration/interpolation errors compared to the noise errors decreases proportionately to at least $N^{2½}$ for the "1(NC), 1st order" block, and by even more than this for the more complex blocks. The exponent 2½ here derives from the N^{-3}

relationship between sample rate and interpolation error, times the √N relationship between sample rate and noise error, for the "1(NC), 1st order" block.

3.13.1   Selection process for algorithm selection

Gathering all the findings and all the algorithm CPU requirements from sections 3.6 through to this point, it is now possible to create a selection process, which provides a clear path to a decision about which set of algorithms to use in a particular application. To select the most appropriate algorithm, these four questions need to be answered:-

1.   Is the sample rate less than 16 samples per cycle? (800 Sa/s @ 50 Hz)?

2.   Is the effect of noise "small"? Guidelines are that noise should be <0.001pu @ 500 Sa/s, or < 0.0002pu at 1000 Sa/s, or <$0.001/(R/500)^{2\frac{1}{2}}$ where R is some other sample rate. The relationship with exponent 2½ was deduced amongst the findings in section 3.13. When noise is too large, the gain of the cancellation algorithms and 2nd order interpolation is lost. The noise is set by the quality of the instrumentation hardware, and is generally a fixed quantity.

3.   Is ½ cycle measurement speed required?

4.   Is a 1-cycle measurement speed required? AND is the harmonic content sometimes small? If the harmonic content is always bad (verging on BS EN 50160 violations) then the answer to this question is "No". However, even within microgrids the harmonic content may be quite low, even if this is not true always. In this case, the answer to this question may be "Yes"

Now, use the flowchart in Figure 3-78 with the answers to questions 1-4 and complete the process.

The computational effort required to carry out the various algorithms varies by algorithm. For a full analysis, the reader should refer to sections 3.6 to 3.12 which describe the algorithms. A simplistic comparison of the computational effort can be made by comparing the numbers of "hard" maths functions required for each algorithm. In Table 3-3 below, the functions which are identified as "hard" are listed. These present significantly larger burdens on a CPU than multiplication/addition/subtraction processes (see also Appendix G).
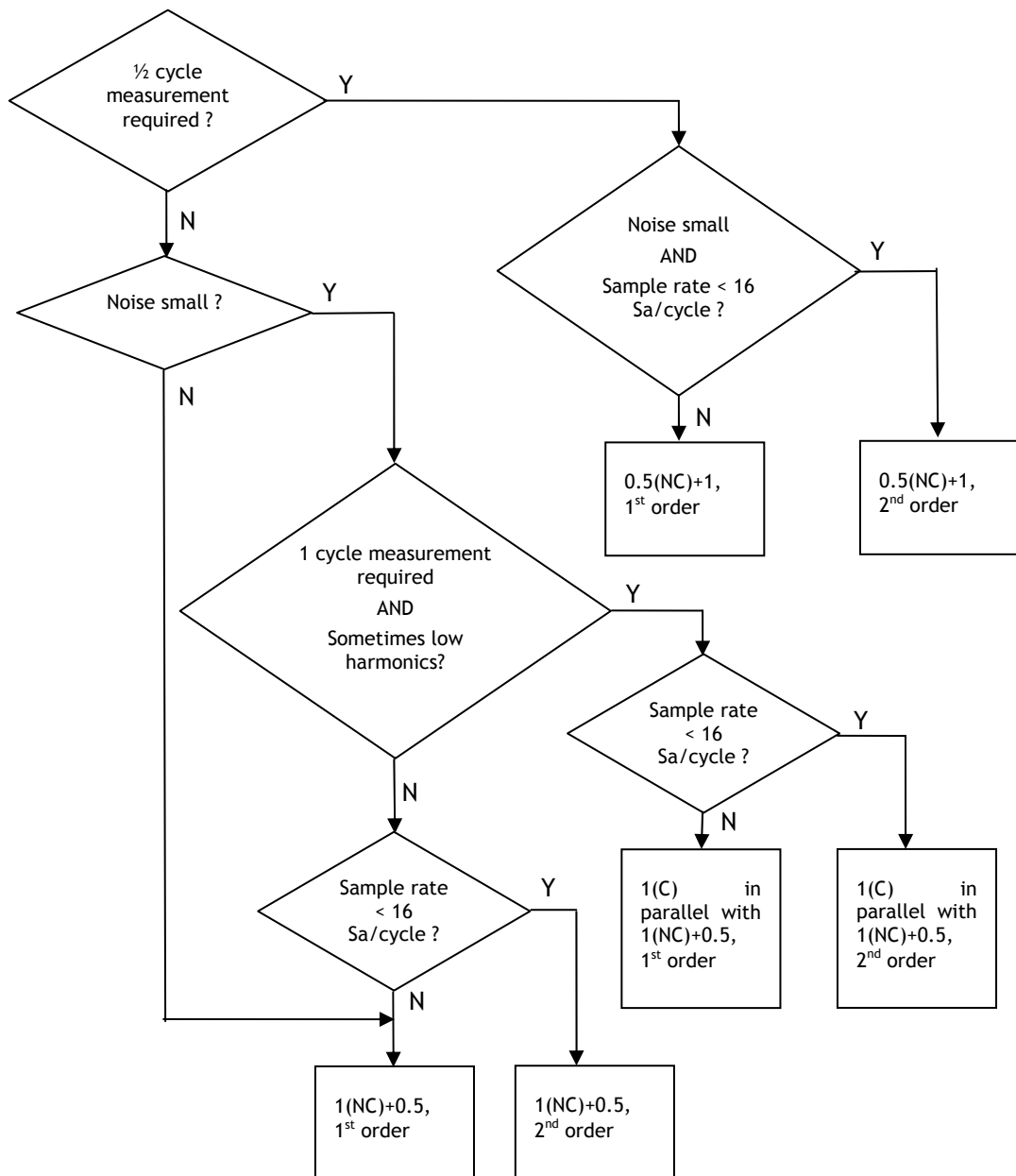
Figure 3-78 : Selection flowchart for measurement algorithm selection

sine/cosine/tangent
arcsin/arccos
atan/atan2
sqrt
x^y (y not integer)
e^x, ln(x), 10^x, $\log_{10}(x)$

Table 3-3 : "hard" maths functions

In Table 3-4, the number of these "hard" maths functions required for each algorithm (per measurement channel) is listed. The two columns describe the number of function evaluations required, both with all-harmonic RMS & THD calculations, and also for a stripped-down algorithm with the all-harmonic RMS & THD calculations removed. Note that all 2nd order algorithms have the same number of "hard" math functions as their 1st order counterparts, but that there are additional CPU operations required as described in section 3.2.2, so the 2nd order blocks should only be used where there is definite benefit.

For all the algorithms, the "Part A" data requires a "hard" math function count of 2 (sine and cosine) (see section 3.6). This is unavoidable, is common to all proposed measurement systems, and can be re-used for many measurement blocks at the same frequency. Therefore, it is not included in the counts of "hard" math functions below.

Algorithm	"Hard" math function count per measurement channel. including all-harmonic RMS and THD calculations	"Hard" math function count per measurement channel. without all-harmonic RMS and THD calculations
1(C) in parallel with 1(NC)+0.5, 2nd order	8	6
1(C) in parallel with 1(NC)+0.5, 1st order	8	6
1(NC)+0.5, 2nd order	4	2
1(NC)+0.5, 1st order	4	2
0.5(NC)+1, 2nd order	4	2
0.5(NC)+1, 1st order	4	2

Table 3-4 : "Hard" maths functions for different measurement algorithms

3.14 Final selection of algorithms for further development and testing in this thesis

The applications most of interest within the scope of this thesis are those with the following properties:-

- Voltage measurements (amplitude, phase, RMS, THD, frequency), at 500 Sa/s or slightly above, with measurement latencies of 1.5 to 5 cycles, in an environment where the noise level is currently ≈0.005pu (significant), and harmonic contamination may be (but is not always) high. From Figure 3-78, the appropriate algorithm is therefore "1(NC)+0.5, 1st order". With a lower noise level, the "1(NC)+0.5, 2nd order" algorithm might also become appropriate.

Therefore, the two most relevant blocks "1(NC)+0.5, 1st order" and "1(NC)+0.5, 2nd order" are explored further in the next section, and are also used as the prime building blocks within the frequency measurement algorithms of section 5. To remind the reader, these classifications translate as:-

- "1(NC)+0.5, 1st order" = A Fourier transformation over exactly 1 cycle, with no 2nd-harmonic cancellation, with a subsequent ½ cycle averaging of the results, using 1st order integration and interpolation techniques throughout.

- "1(NC)+0.5, 2nd order" = ditto, but using 2nd order integration and interpolation techniques throughout.

The next chapter goes on to deal with two further error mechanisms:-

- Errors due to inaccurate estimates of signal frequency

- Errors due to aliased harmonics

3.15   References for chapter 3

Aghazadeh, R. et al. (2005). New technique for frequency and amplitude estimation of power system signals. IEE Proceedings on generation, transmission and distribution, vol. 152, no. 3, pp 435-440.

Dash, P. et. al. (2000). An extended complex Kalman filter for frequency measurement of distorted signals. IEEE transactions on instrumentation and measurement, vol.49 issue 4. pp 746-53.

Johns, A. & Salman, S. (1995). Digital protection for power systems. IEE Power Series 15, ISBN 086341303X.

Jovcic, D. (2003). Phase locked loop system for FACTS. IEEE transactions on power systems vol.18 issue 3. pp 1116-24.

Kuhlmann, V. et al. (2007). Effects of sampling rate and ADC width on the accuracy of amplitude and phase measurements in power-quality monitoring. IEEE Transactions on power delivery, vol. 22 no. 2, pp 758-764.

Lin, T. & Domijan, A. (2005). On power quality indices and real time measurement. IEEE Transactions on power delivery, vol. 20 no. 4, pp 2552-2562.

Moore, P., Carranza, R. & Johns, A. (1996a). Model system tests on a new numeric method of power system frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 2, pp 696-701.

Moore, P., Allmeling, J. & Johns, A. (1996b). Frequency relaying based on instantaneous frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 4, pp 1737-1747.

Petrovic, P., Marjanovic, S. & Stevanovic, M. (2000). New algorithm for measuring 50/60 Hz AC values based on the usage of slow A/D converters. IEEE transactions on instrumentation and measurement, vol.49 issue 1. pp 166-171.

Welch., G & Bishop, G. (2001). An introduction to the Kalman filter. Online at: <http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf> [Accessed 11/03/2008]

4 Higher order harmonics and aliasing, and the effects of frequency measurement error

This chapter follows directly from the work of chapter 3. The topics covered are:

- Assessment of the algorithm performance due to frequency measurement error.

- Assessment of the algorithm performance against aliased harmonics, accounting for the actual performance of the anti-aliasing filters which do not remove all signal components above the nyquist frequency (see section 3.4.1).

- The response of the algorithm outputs under the influence of aliased harmonics is found to be a ripple, with a frequency which may be sub-fundamental. For measurements which can afford a longer measurement time than 1½ cycles, a novel ripple-removal filter is designed, based yet again on the exact-time-period averaging blocks.

- Overall findings and guideline error-magnitude error levels are presented.

- To meet the toughest amplitude measurement ripple specifications at waveforms of up to 28-53% THD_V, with the main algorithm running at 10 samples per cycle (500 Sa/s), a digital oversampling at 3kHz and a simple 6-tap FIR filter is required at the front end. This filter is designed and tested with the whole algorithm to verify the improved performance. Methods to achieve this oversampling inside economical microcontroller solutions are described.

4.1 The effect of frequency measurement error

In the previous chapter, the measurement block inputs were fed with the test input signal, plus the actual signal frequency. This is possible in simulations where the input signal is synthesised directly. In real scenarios the frequency must be measured, and this measure of frequency will always be in error by some amount. The worst frequency error which would be expected from the measurement algorithms proposed in chapter 5, in the microgrid context, was defined by the requirements of Table 2-14. This worst reasonably expected error is caused by a 5-cycle response measurement, with a ROCOF rate of 10 Hz/s; this equates to 10*5/50 = 1 Hz for a 50 Hz system. To examine the effect of this magnitude of error on the amplitude/phase measurements, the simulation from the previous section 3.13 can be re-used. The only difference is that now a ±1 Hz frequency error is introduced onto the estimate of frequency passed into the measurement algorithms. The resulting errors are shown below in Figure 4-1, compared to the errors resulting from pure sinusoid inputs and 0.005pu RMS noise (without frequency errors, from section 3.13) to give context.

Before analysing the chart, the results can be qualitatively described as follows:

- The result of the frequency measurement error is that the initial Fourier transformation block exhibits ripple. For the 1-cycle base Fourier measurement, this ripple is at exactly 2x the actual signal frequency. It was expected that this ripple might show up at a combination of mixed frequencies such as N times the frequency estimate ± M times the actual signal frequency. However, careful analysis of the ripple using Fourier transformation confirms that the ripple is almost entirely confined to a single frequency at 2x the actual signal frequency (see Table 4-1), just as was the ripple due to interpolation. If the initial Fourier measurement base block is of ½-cycle duration, then the ripple is at the fundamental. Thus, the "1(NC)", "1(C)", and "0.5(NC)" base measurement block outputs all exhibit substantial ripple due to frequency measurement error. This effect was also noted by Moore (1996b), in a relay which could not adapt well to frequencies off-nominal due to processing limitations. Very small ripples also appear at 0 Hz and integer multiples of the signal frequency, but (somewhat surprisingly) ripples do not appear at, for example, the signal frequency ± the frequency estimate. This is extremely useful since such ripples could have very low frequencies and be hard to remove.

- Due to the post-averaging stages, i.e. the extra ½-cycle averaging after the initial 1-cycle base block, the ripple is mostly removed. This works for both the "1+0.5" cycle and the "0.5+1" cycle systems. However, the ripple removal is not perfect because the ripple removal filter is also fed with the wrong measured frequency.
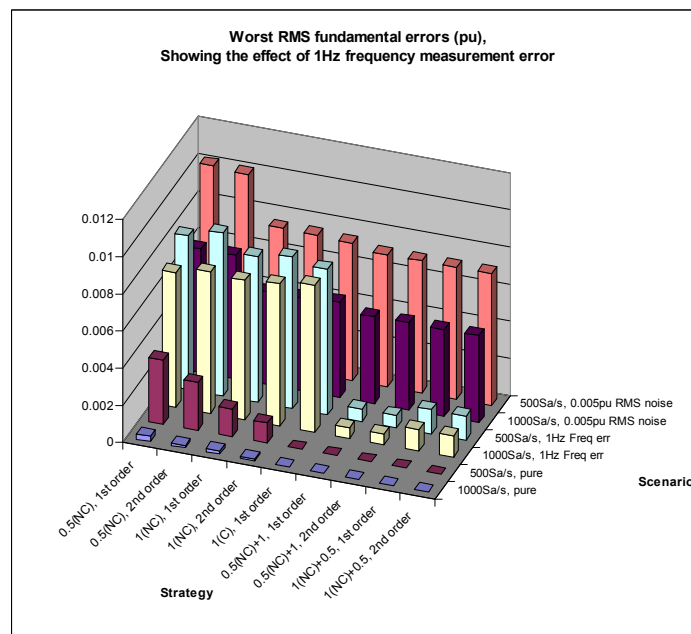
Figure 4-1 : Worst RMS errors on the measurement of fundamental amplitude, due to ±1Hz frequency measurement errors

Sample rate	Signal frequency	Frequency estimate	RMS Ripple magnitude	Ripple frequency
500 Sa/s	50	49	0.00717	100 Hz
500 Sa/s	51	50	0.00603	102 Hz
500 Sa/s	51	51	0.00114	102 Hz
500 Sa/s	52	51	0.00487	104 Hz
1000 Sa/s	52	51	0.00656	104 Hz

Table 4-1 : Ripple from 1-cycle Fourier measurement, due to frequency estimate error

The results show that the error magnitude due to frequency measurement error is not dependent upon the sample rate. The worst error magnitudes, at the outputs of the ½ and 1-cycle latency measurement blocks, are of the same order as the errors due to 0.005pu RMS noise for the 500Sa/s case. For higher sample rates, the error due to frequency measurement error becomes dominant over the effect of the 0.005pu RMS noise. This is because the error due to noise drops as $\sqrt{SampleRate}$ whereas the error due to frequency measurement error stays constant. The worst error for the ½-cycle latency measurement block is 0.008pu RMS, which is acceptable, considering that this will only occur transiently during the fastest ROCOF events.

The worst error at the 1½-cycle outputs is only 0.0012pu RMS, which is of no concern. This shows that the second stages of averaging perform relatively well, even when the wrong frequency estimate is used. Again, this error will only occur transiently during the fastest ROCOF events.

4.2 The effect of aliased harmonic distortion on amplitude measurement accuracy

The sampling process combined with imperfect attenuation of the higher-order harmonics in the anti-aliasing filters can cause the largest measurement errors. These errors arise due to fundamental mathematical properties of the sampled, aliased harmonics and how they interact with the Fourier measurement process. The errors appear as ripples at frequencies which can be anywhere between 0 Hz (DC) and the Nyquist frequency. The worst (dominant) errors are those which appear at or close to DC, because these low frequency ripples are not significantly attenuated by digital averaging stages of 1 or 1½ cycles. For this reason, all of the algorithms proposed thus far react in almost the same way to the aliased harmonics, so it is safe to analyse just the two algorithms "1(NC)+0.5, 1st order" and "1(NC)+0.5, 2nd order" in the next section, and assume that the errors from aliased harmonics will be about the same for all the other algorithms proposed thus far.

To analyse the measurement errors due to aliased harmonics, it is possible to again use the same setup described in section 3.7. However, due to the combination of number of possible frequencies, samples per cycle and harmonic numbers, the required test run to gather all the data would take an unacceptably long time. To address this, the approximate errors are deduced mathematically, and the formulae cross-checked against the simulation to determine accuracy. From this, the most sensible values of samples-per-cycle to use (to attain optimum accuracy) can be much more quickly identified. For these optimal values, the harmonics which cause the worst effects can also be predicted and then analysed further using the full simulation.

One point should be reinforced here. In the error analyses below, if the interfering harmonic is at a frequency which is an integer multiple of the fundamental frequency (which it usually is), then measurement error and ripple will be zero if the harmonic is not aliased (aside from interpolation error). The errors, however, are not zero when the original harmonic is aliased due to sampling. This occurs if the harmonic frequency is above the Nyquist frequency. The harmonic then appears in the digital domain as a signal which is not at an integer multiple of the fundamental frequency.

4.2.1 Theoretical aliasing effects and ripple frequency of the Fourier and RMS measurements due to harmonic contamination

If the sampling frequency of the measurement device is f_s, in Sa/s, then the Nyquist frequency is $f_s/2$ Hz. Any harmonic which appears above this frequency will be aliased upon sampling. The frequency at which the alias appears in the digitally sampled data can be calculated as follows:

Imagine an incoming harmonic at frequency f_h, which is above the nyquist frequency $f_s/2$.

Set $p = \dfrac{f_h}{f_s}$ modulo 1 (modulo being the real-number remainder function)

Now, find the aliased frequency f_a by

$$f_a = f_s(1-p) \qquad \text{if } p>0.5$$

$$f_a = f_s p \qquad \text{otherwise}$$

and $\omega_a = 2\pi f_a$

(4.1)

An example of this effect is if f_s =500 Sa/s, with a fundamental f_f at 50 Hz (10 samples per

cycle), then the 11th harmonic at f_h=550Hz will result in p=550/500 modulo 1 which equals 0.1, resulting in an aliased frequency f_a of 500*0.1=50 Hz which exactly overlies the fundamental.

The expected ripple frequency of the Fourier and RMS measurement ripples described in sections 4.2.2 and 4.2.3 will be:-

$$f_{ripple} = \left| f_f - f_a \right|$$ where f_a is the aliased frequency resulting from the actual harmonic at frequency f_h, and f_f is the fundamental frequency of the main waveform.

As in the example above, when f_f=f_a then f_{ripple} is 0. In this special case, a ripple is not seen but a constant (DC) error appears on any measurements.

4.2.2   Theoretical effect of harmonic and inter-harmonic distortion on measurement of the Fourier fundamental.

The Fourier measurement of the fundamental component is made by correlating the measured waveform against a sine/cosine pair at the fundamental frequency f_f (ω_f =2πf_f). In the analysis, we only need to consider the sin component since the phase can be set such that the cosine component is zero due to symmetry.
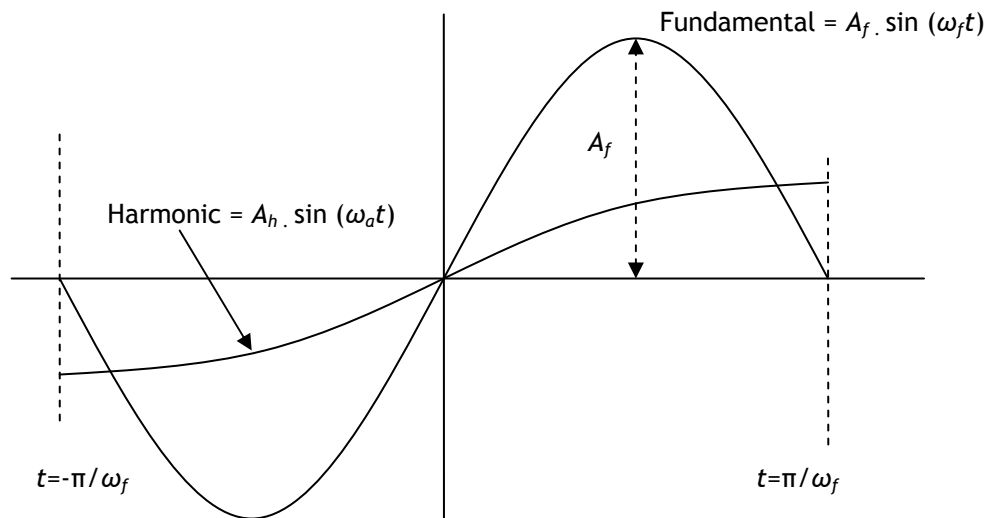
Figure 4-2 : Derivation of the formula for fundamental amplitude errors due to a harmonic, sub-harmonic or inter-harmonic

The fundamental component magnitude (at ω_f) is calculated as

$$A_{f\_meas} = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} y(t) \cdot \sin(\omega_f t) \cdot dt$$

$$T = \frac{2\pi \cdot N}{\omega_f}$$

Where $y(t)$ is the measured waveform and N is the integer number of whole cycles. N is normally 1, but can be more to reduce noise at the expense of latency.

It is useful here to define K, since this allows simplification of the subsequent equations.

$$K = \frac{\omega_f}{N\pi} = \frac{2}{T}$$

(4.2)

The worst case for harmonic addition is when the aliased harmonic is most correlated with the fundamental, with the zero crossings aligned as shown in Figure 4-2. In this instance, the contribution to A_{f_meas} due to the harmonic at amplitude A_h, with (aliased) frequency f_a ($\omega_a = 2\pi f_a$), will be:

$$A_{f\_meas} = K \int_{-\frac{1}{K}}^{\frac{1}{K}} A_h \cdot \sin(\omega_a t) \cdot \sin(\omega_f t) \cdot dt$$

This can be solved by using the identity sin(a).sin(b) = ½(cos(a-b)-cos(a+b)) to give

$$A_{f\_meas} = K \cdot A_h \cdot \left(\left(\frac{\sin\left(\frac{\omega_a - \omega_f}{K} \right)}{(\omega_a - \omega_f)} \right) - \left(\frac{\sin\left(\frac{\omega_a + \omega_f}{K} \right)}{\omega_a + \omega_f} \right) \right)$$

This expression may be re-expressed in terms of frequency f instead of ω as follows:

$$K_{freq} = \frac{f_f}{N\pi} = \frac{K}{2\pi}$$

$$A_{f\_meas} = K_{freq} \cdot A_h \cdot \left(\left(\frac{\sin\left(\frac{f_a - f_f}{K_{freq}} \right)}{(f_a - f_f)} \right) - \left(\frac{\sin\left(\frac{f_a + f_f}{K_{freq}} \right)}{f_a + f_f} \right) \right)$$

The resulting per-unit error on the measurement of fundamental amplitude can be calculated by setting A_h to the per-unit amplitude of the interfering harmonic (e.g. 0.05 for 5%), and calculating the absolute value of A_{f_meas}.

4.2.3 Theoretical effect of harmonic and inter-harmonic distortion on measurement of the all-harmonic RMS value.

The all-harmonic RMS measurement of the input waveform is made by a standard RMS process on the waveform. The worst cases are where the interfering harmonic peaks (or troughs) coincide with the peaks of the fundamental waveform.
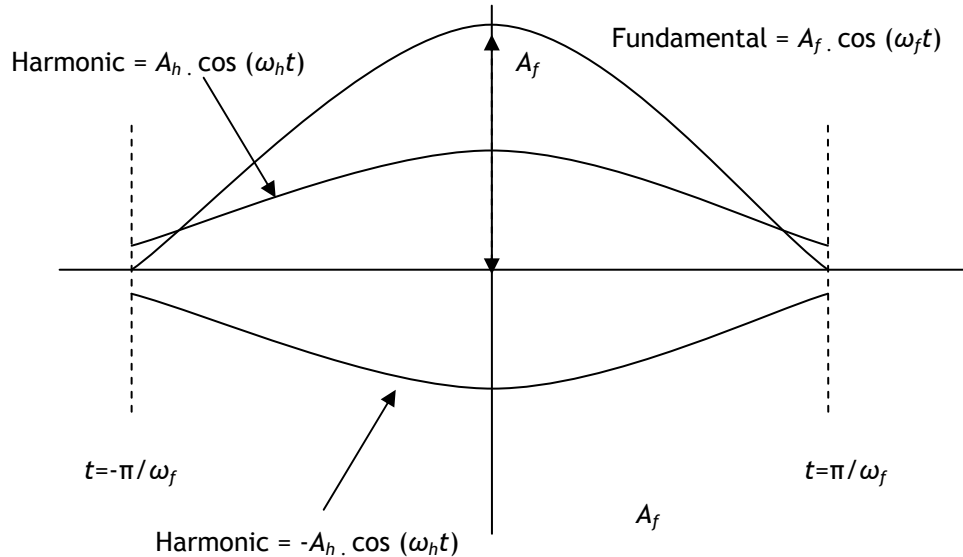
Figure 4-3 : Derivation of the formula for fundamental amplitude errors due to a harmonic, sub-harmonic or inter-harmonic

The RMS magnitude is calculated as A_{RMS_meas}, which here we wish to give the peak value of the real waveform; i.e. we would wish $A_{RMS\_meas}=A_f$ if there were no harmonics present. Thus, there is a factor of $\sqrt{2}$ in the equation below, which does not appear in (3.7). This accounts for the transformation from RMS to peak values.

$$A_{RMS\_meas} = \sqrt{2} \sqrt{\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} y^2(t) \cdot dt} = \sqrt{\frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} y^2(t) \cdot dt}$$

$$T = \frac{2\pi \cdot N}{\omega_f}$$

Where $y(t)$ is the measured waveform, and N is the integer number of whole cycles. N is normally 1, but can be more to reduce noise at the expense of latency.

It is useful here to define K, since this allows simplification of the subsequent equations.

$$K = \frac{\omega_f}{N\pi} = \frac{2}{T}$$

(4.3)

The worst case for harmonic addition is when the peaks and troughs of the fundamental and the interfering harmonic are aligned as shown in Figure 4-3. The alignment of peaks is considered below. The alignment of a peak with a trough can be considered later with the same equations but by using a negative value of A_h. In this instance, the final answer A_{RMS_meas} due to the fundamental at f_f, A_f plus the harmonic at amplitude A_h, (aliased) frequency f_a ($\omega_a = 2\pi f_a$), will be:

$$A_{RMS\_meas} = \sqrt{K \int_{-\frac{1}{K}}^{\frac{1}{K}} \left(A_f \cdot \cos(\omega_f t) + A_h \cdot \cos(\omega_a t) \right)^2 \cdot dt}$$

This can be solved by expanding and using the identities $\cos^2(a) = \frac{1}{2}(1+\cos(2a))$ & $\cos(a).\cos(b) = \frac{1}{2}(\cos(a+b)+\cos(a-b))$ to give

$$A_{RMS\_meas} = \sqrt{A_f^2 + A_h^2\left(1 + \frac{K\sin\left(\frac{2\omega_a}{K}\right)}{2\omega_a}\right) + 2KA_f A_h\left(\left(\frac{\sin\left(\frac{\omega_f + \omega_a}{K}\right)}{\omega_f + \omega_a}\right) + \left(\frac{\sin\left(\frac{\omega_f - \omega_a}{K}\right)}{\omega_f - \omega_a}\right)\right)} \text{ Th}$$

is expression may be re-expressed in terms of frequency F instead of ω as follows:

$$K_{freq} = \frac{f_f}{N\pi} = \frac{K}{2\pi}$$

$$A_{RMS\_meas} = \sqrt{A_f^2 + A_h^2\left(1 + \frac{K_{freq}\sin\left(\frac{2f_a}{K_{freq}}\right)}{2f_a}\right) + 2K_{freq}A_f A_h\left(\left(\frac{\sin\left(\frac{f_f + f_a}{K_{freq}}\right)}{f_f + f_a}\right) + \left(\frac{\sin\left(\frac{f_f - f_a}{K_{freq}}\right)}{f_f - f_a}\right)\right)}$$

The per-unit error can be calculated by setting A_f to 1, and A_h to the per-unit amplitude of the interfering harmonic (e.g. 0.05 for 5%), and calculating the value of A_{RMS_meas}. The error is the absolute difference between of A_{RMS_meas} and the actual RMS value which can be calculated as

$$A_{RMS\_actual} = \sqrt{A_f^2 + A_h^2} = \sqrt{1 + A_h^2} \text{ when } A_f \text{ is set to 1.}$$

The per-unit RMS errors are thus

$$A_{RMS\_error} = \left| \sqrt{(1 + A_h^2)} - A_{RMS\_meas} \right| \text{ with } A_f \text{ set to 1.}$$

Note that for each harmonic frequency and amplitude, A_h should be set to both the +ve and –ve value for the harmonic amplitude of interest, and the worst case result taken.

4.2.4 Comparison of theoretical to simulated results for harmonic contamination

Using the formulae derived in sections 4.2.1 through 4.2.3, the following 3 predictions were made (Figure 4-4 to Figure 4-6). The scenario here is:-

- Nominal frequency 50Hz

- 10 Samples per cycle (500 Sa/s)

- 11th harmonic added, at 0.035pu (the BS EN 50160 (BSI, 2000) specification). The harmonic is not considered in this example to be attenuated by any low-pass anti-alias filter.

- Actual input frequency varied from 45 to 55Hz

- Measurements made over 1 whole cycle
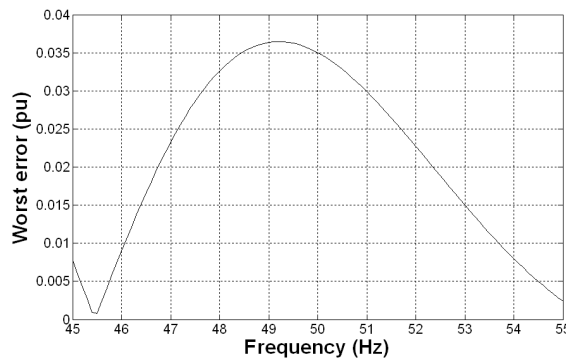
Figure 4-4 : Prediction of errors on the Fourier fundamental measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz
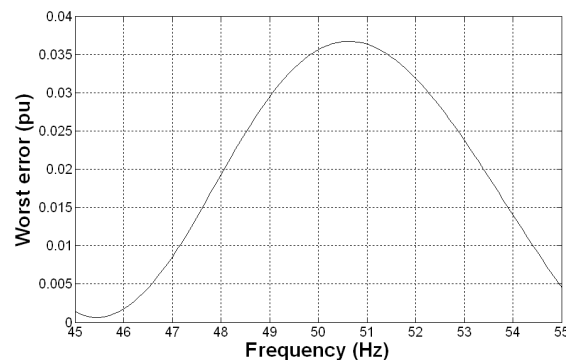
Figure 4-5 : Prediction of errors on the all-harmonic RMS measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz

163
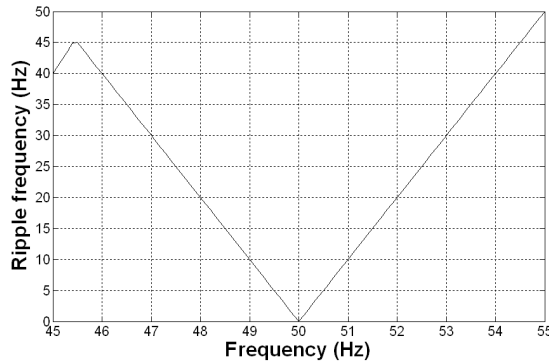
Figure 4-6 : Prediction of ripple frequencies on measurements at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz

A set of simulations using the model/test script described in section 3.7 produces the following graphs. Both 1st and 2nd order interpolation/integration algorithm results (see sections 3.2.2 and 3.7) are shown in the graphs below. They produce very similar results in the scenarios presented here, because the potential errors due to aliased harmonics are larger than the interpolation & integration errors. The ripple frequencies (Figure 4-9) match those predicted (Figure 4-6) exactly. The magnitude and shape of the error plots matches well, but not exactly. This is due to additional properties of the actual measurement algorithms which are not predicted by the theoretical alias analysis. The most important of these properties are the integration and interpolation errors which interact with the interfering harmonic. These errors have been thoroughly investigated in section 3.7. Although the match of Figure 4-4 & Figure 4-5 to Figure 4-7 & Figure 4-8 is thus not perfect, it is good enough to justify use of the mathematical models to predict the approximate magnitudes of errors, and for what input frequencies they will appear at different sample rates. This is extremely useful as a design tool, as will be seen in section 4.2.5.
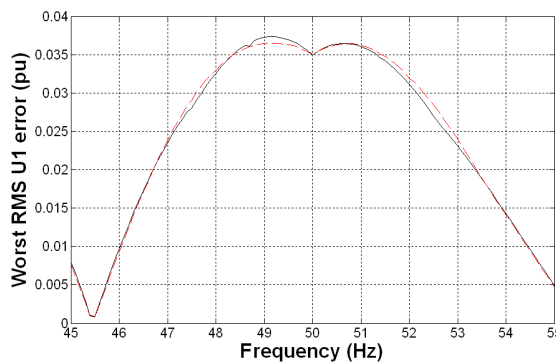
Figure 4-7 : Simulation of errors on the Fourier fundamental measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz. 1st order (solid) and 2nd order (red dashes) methods

164
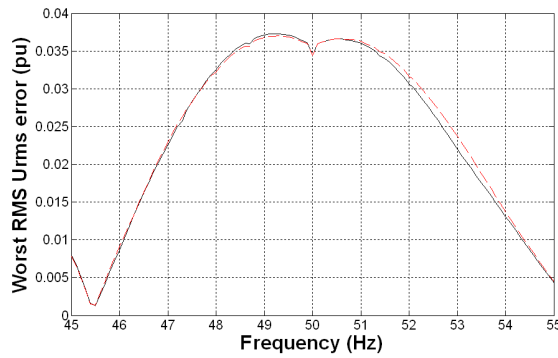
Figure 4-8 : Simulation of errors on the all-harmonic RMS measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz. 1st order (solid) and 2nd order (red dashes) methods
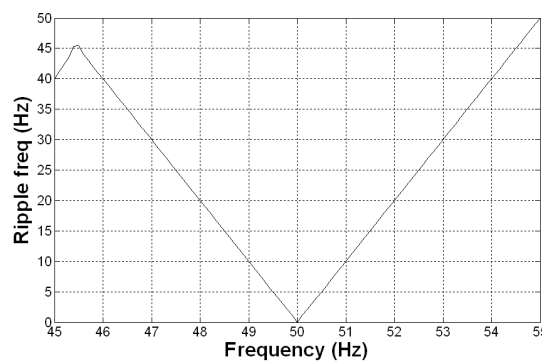
Figure 4-9 : Simulation of ripple frequencies on measurements at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz.

To double-check the quality of the agreement between prediction and simulation, the prediction software was used to predict that the 23rd harmonic would cause an appreciable effect at 26 samples per cycle. This is just one interesting combination picked from a choice of many. With the 23rd harmonic at the 1.5% level as per BS EN 50160 (BSI, 2000), the predictions for errors are shown in Figure 4-10 and Figure 4-11.
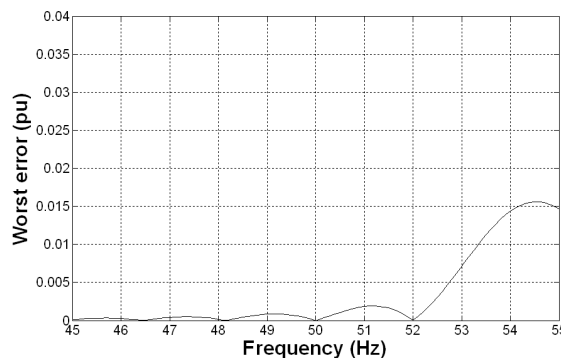
Figure 4-10 : Prediction of errors on the Fourier fundamental measurement at 1300Sa/s due to 23rd harmonic @ 1.5%, for input frequencies in the range 45-55Hz

165
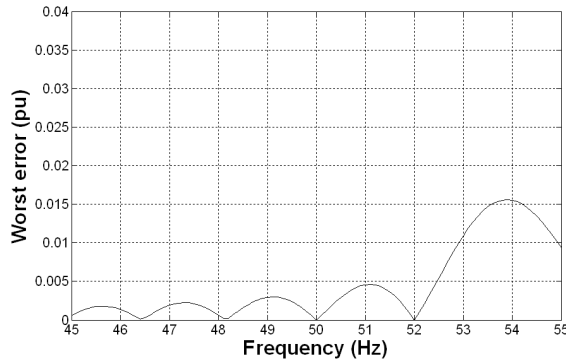
Figure 4-11 : Prediction of errors on the all-harmonic RMS measurement at 1300Sa/s due to 23rd harmonic @ 1.5%, for input frequencies in the range 45-55Hz

The simulated results are shown below in Figure 4-12 and Figure 4-13. The agreement between prediction (Figure 4-10 & Figure 4-11) and actual simulation is again good. This provides more evidence that the prediction tools (which execute much more quickly than the simulation sets) are valid. This means that the prediction tools can be used to predict error levels for given scenarios of sample rate and harmonic contamination. This is carried out in section 4.2.5.
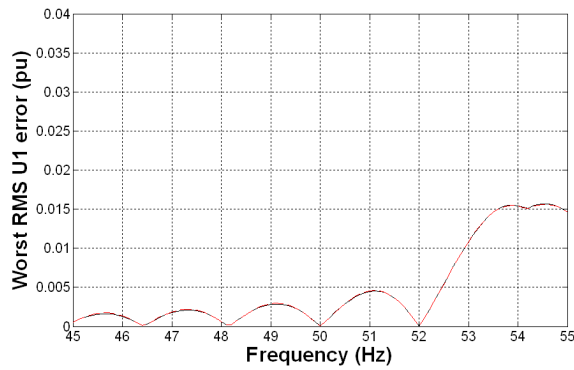
Figure 4-12 : Simulation of errors on the Fourier fundamental measurement at 1300Sa/s due to 23rd harmonic @ 1.5%, for input frequencies in the range 45-55Hz. 1st order (solid) and 2nd order (red dashes) methods
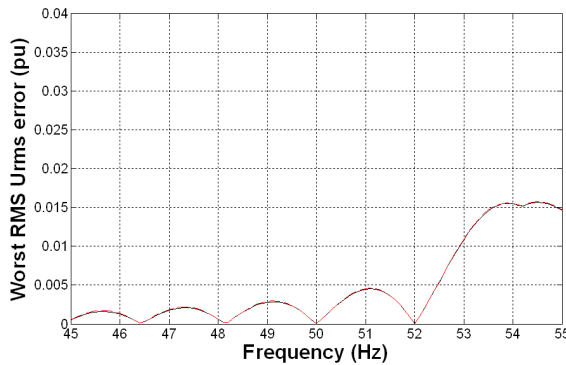
Figure 4-13 : Simulation of errors on the all-harmonic RMS measurement at 1300Sa/s due to 23rd harmonic @ 1.5%, for input frequencies in the range 45-55Hz. 1st order (solid) and 2nd order (red dashes) methods

4.2.5   Prediction of errors at different sample rates

Having established a set of tools to predict the errors due to harmonic contamination, these tools can be used to quickly explore the relationship between these errors and the sampling rate. Before this is done, a final touch is to add models of the low-pass anti-aliasing filter stages described in section 3.4. The tools can then be repeatedly executed using a MATLAB script, to examine the following space of scenarios:-

- All values of samples per cycle from 10 to 30

- All harmonics from 2 to 40

- Individual harmonic amplitudes set to either a multiple of the BS EN 50160 (BSI, 2000) levels (see Figure 2-2, using values of 0.5% for all harmonics above the 24th), or to the worst-case microgrid harmonic voltage amplitudes outlined in section 2.7.2

- All values of input frequency in 0.1Hz steps, from 45 to 55 Hz

The entire resulting dataset can be summarised and cut in a number of different ways to show the effects of the harmonics and the different sample rates.

4.2.5.1   Errors due to aliased and un-aliased harmonics at 2x the BS EN 50160 levels

In the plots shown below, the point plotted for each value of samples-per-cycle shows the maximum error which we would expect to see by applying all the harmonics 2 to 40 at levels twice those of BS EN 50160 (BSI, 2000), simultaneously. This corresponds to a THD of 22.8% and would violate the overall 8% THD specification of BS EN 50160 by a factor of more than 2. At each value of sample-per-cycle, the values of the errors are calculated by the following process:

- Select each of the input frequency values in turn from the range 45-55 Hz

- calculate the RSS (Root sum of squares) of all the errors due to every even harmonic 2-40 at this input frequency, (even harmonics are likely to be uncorrelated)

- then linearly add the errors due to every odd harmonic (odd harmonics are likely to be correlated due to saturation and distortion effects)

- repeat for all input frequencies at this value of samples-per-cycle

- select the worst value of RSS error found

Note that this treatment by RSS of the even harmonics plus linear summing of the correlated odd harmonics matches the rationale used to generate the expected microgrid harmonic levels in section 2.7.2.

First, the predicted errors on the Fourier fundamental calculation are shown in Figure 4-14, and then the predicted errors on the all-harmonic RMS measurement in Figure 4-15.
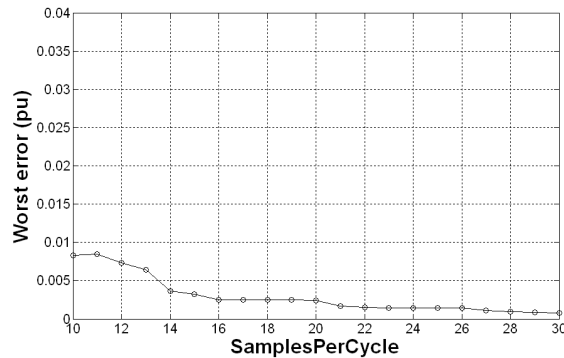
Figure 4-14 : Prediction of worst overall errors on Fourier fundamental measurements, due to 2x BS EN 50160 level harmonics 2-40 applied all at once
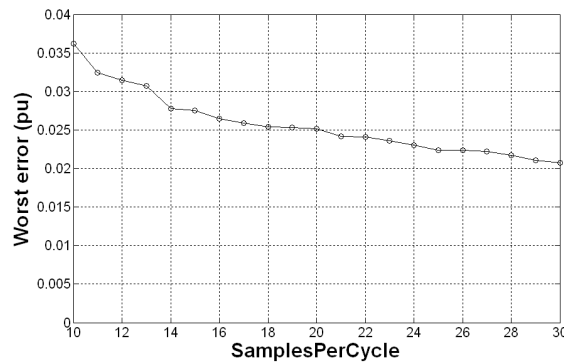
Figure 4-15 : Prediction of worst overall errors on all-harmonic RMS measurements, due to 2x BS EN 50160 level harmonics 2-40 applied all at once

The worst expected errors on the Fourier fundamental measurement are <0.01pu for all values of sample-per-sample. The all-harmonic RMS errors are considerably larger, up to 0.04 pu at 10 samples per cycle. The larger all-harmonic RMS errors, particularly at the low sample rates, are caused by the attenuation of the harmonic content by the anti-aliasing filters. Even at 30 samples per cycle, the filter cut-off frequency is only 250 Hz (1/3rd of the Nyquist frequency which is ½ of 50*30). Therefore all harmonics of 5th and above are well attenuated and are not then measurable by the all-harmonic RMS algorithm. The all-harmonic RMS measurement tends therefore to give a lower result than the actual waveform RMS value. This will also cause an incorrectly low THD readout. The required sample rate to record an accurate all-harmonic RMS measurement for everything up to the 40th harmonic would then be approximately defined by being able to set the anti-aliasing filter cutoff frequency to 40*50Hz = 2kHz. This would imply a Nyquist frequency of about 3x this amount (6kHz) and therefore a sample rate of about 12 kSa/s,

or 240 samples per cycle.

This can be verified by a repeat analysis over a wider range of samples per cycle, which shows that the all-harmonic RMS measurements do become accurate to the 0.0025pu level for 2x BS EN 50160 (BSI, 2000) harmonic inputs at around the 240 sample per cycle rate, with a low-pass filter cut-off frequency of 2kHz (Figure 4-16).
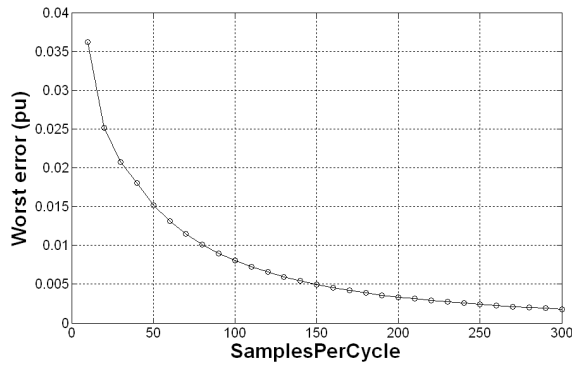
Figure 4-16 : Prediction of worst overall errors on all-harmonic RMS measurements, due to 2x BS EN 50160 level harmonics 2-40 applied all at once. 10-300 Sa/cycle

Even at this sample rate and error level, by equation (3.8), the THD error due to a 0.0025pu error on the RMS measurement would be 7%. To achieve <1% accuracy in the THD measurement of harmonics up to the 40th, the sample rate required is about 2000 samples per cycle (100 kSa/s) to give an RMS error of 0.00005pu (Figure 4-17). An alternative would be to measure each of the harmonic amplitudes separately using Fourier fundamental measurements. This would require 40 Fourier measurements to measure up to the 40th harmonic (or an FFT process to measure all concurrently), with a sample rate of approximately 10*40=400 Samples per cycle, 20 kSa/s (A sample rate of 128 Samples per cycle is used in just this way in Kuhlmann (2007)). Even then, the answer would only be accurate upon the assumption that no inter-harmonics existed. Since modern power-electronic devices inject harmonics at frequencies which are locked to quartz clocks and not the system frequency, such an assumption is invalid. The kinds of sample rate and required processor speed mentioned above are outside the aim and scope of this document.

169
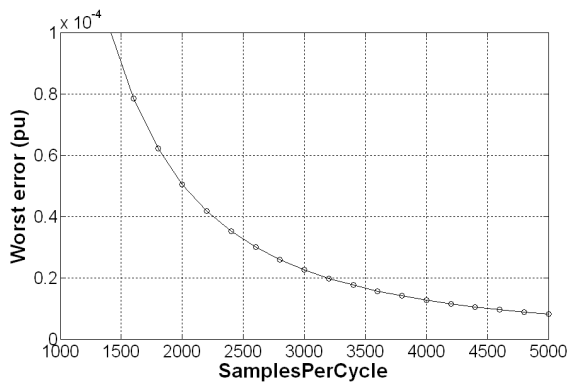
Figure 4-17 : Prediction of worst overall errors on all-harmonic RMS measurements, due to 2x BS EN 50160 level harmonics 2-40 applied all at once. 200-5000 Sa/cycle

Sample rates as low as 10 samples-per-cycle are therefore viable within a microgrid scenario, but only the measurements of Fourier fundamental will be robust and accurate. With high levels of harmonics the all-harmonic RMS measurements might be in error by up to 4% under the scenario described, which is an unacceptably large error. THD errors are even larger.

4.2.5.2 Errors due to aliased and un-aliased harmonics at worst microgrid levels

Finally, the predicted error analysis is repeated but for the worst case microgrid harmonic content scenario described in section 2.7.2. This waveform has a THD_V of 53%. The errors on the Fourier fundamental measurement at 10 samples per cycle are ≈0.02pu which is quite large (Figure 4-18). The errors on the all-harmonic RMS measurement are much larger, of the order of 0.15pu, which makes the RMS measurements totally unusable (Figure 4-19).
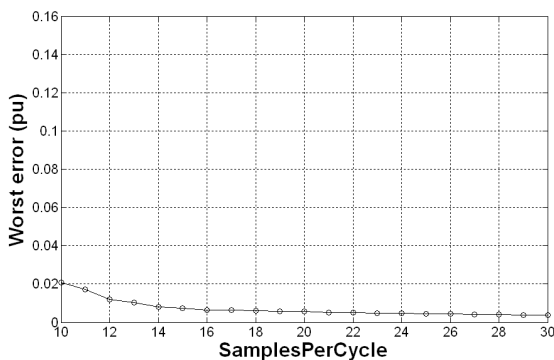
Figure 4-18 : Prediction of worst overall errors on Fourier fundamental measurements, due to worst case microgrid harmonics 2-40 applied all at once

170
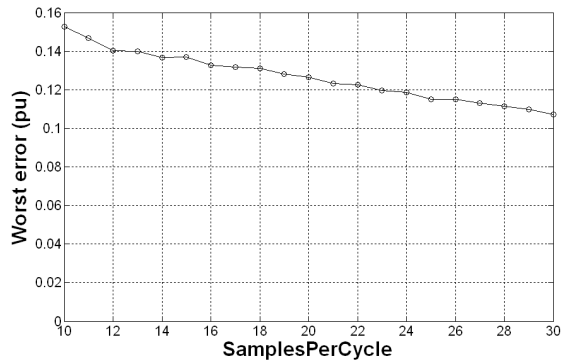
Figure 4-19 : Prediction of worst overall errors on all-harmonic RMS measurements, due to worst case microgrid harmonics 2-40 applied all at once

4.2.6   Validation of the predicted performance at 10Sa/cycle using simulation

The results of section 4.2.5 can be compared to results obtained using the discrete-time simulation model described in section 3.7, this time using waveforms contaminated by harmonics which become aliased (whereas in section 3.7 the waveforms contained no harmonics which were aliased). When this simulation is executed using waveforms containing the worst-case microgrid harmonic levels from section 2.7.2, the results presented below are obtained (Figure 4-20 to Figure 4-23). In these results, the effects of the anti-aliasing filters, noise at 0.005pu RMS, ADC quantisation and DC block have all been included. The measurement is a single 1-cycle Fourier measurement block, without any post-averaging. This is representative for the 1½ cycle algorithms too, since the largest magnitude errors due to aliasing tend to have the lowest ripple frequencies which are not effectively attenuated by the short ½-cycle post-averaging stages in the 1½-cycle algorithms.

The results of Figure 4-20 and Figure 4-22 show good agreement with the predicted errors in Figure 4-18 and Figure 4-19 since the harmonic content at these levels is by far the biggest contributor to error. Due to the levels of harmonics, noise, and ADC quantisation, the 2nd order algorithm does not produce any better results than the 1st order algorithm, but the errors are also no worse. The THD error is as high as 50%!
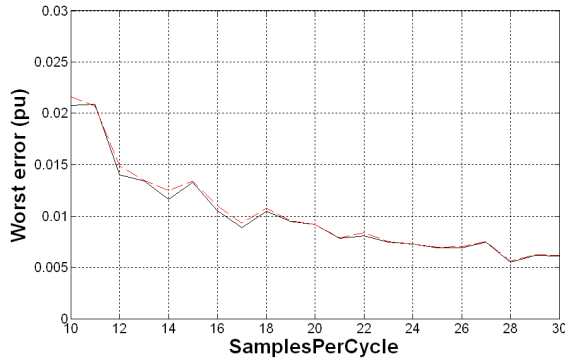
Figure 4-20 : Simulation of worst overall errors on Fourier fundamental measurements, due to worst case microgrid harmonics 2-40 applied all at once, plus noise and ADC quantisation. 1st order (solid) and 2nd order (red dashes) methods
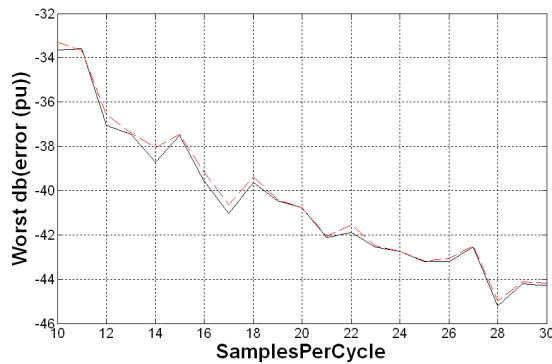
Figure 4-21 : Simulation of worst overall dB(pu errors) on Fourier fundamental measurements, due to worst case microgrid harmonics 2-40 applied all at once, plus noise and ADC quantisation. 1st order (solid) and 2nd order (red dashes) methods

By Figure 4-21, the rate of decrease of errors in approximately 10dB (a linear factor of 3) for a 3-fold increase in sample rate. This means that error is approximately proportional to SampleRate^{-1}.
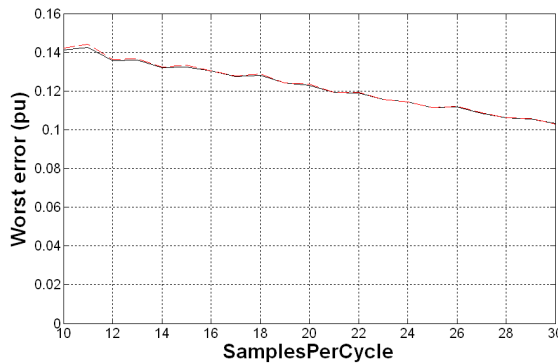
Figure 4-22 : Simulation of worst overall errors on all-harmonic RMS measurements, due to worst case microgrid harmonics 2-40 applied all at once, plus noise and ADC quantisation. 1st order (solid) and 2nd order (red dashes) methods
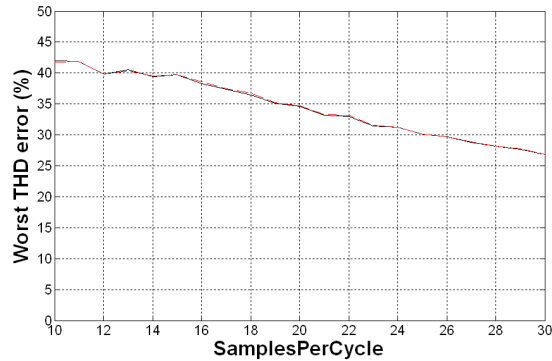
Figure 4-23 : Simulation of worst overall errors on THD measurements, due to worst case microgrid harmonics 2-40 applied all at once, plus noise and ADC quantisation. 1st order (solid) and 2nd order (red dashes) methods

4.2.7   Findings from this section

- The effect of harmonic contributions and aliasing effects can be effectively predicted using the tools generated in this section.

- Sample rates as low as 10 samples per cycle will give errors less than about 0.01pu on the Fourier fundamental measurements, for input waveforms containing harmonics at twice the allowed levels for individual harmonics specified by BS EN 50160, to a total of 22.8% THD. For the worst-case microgrid scenario (53% THD), the worst error is ~0.02pu.

- The errors induced onto the all-harmonic RMS measurements are much larger due to the attenuation of the harmonics by the anti-aliasing filters. Only by raising sample rates to 240 samples per cycle will the all-harmonic RMS measurement become accurate to the 0.0025pu level, for 2x BS EN 50160 harmonic levels. Sample rate needs to be raised to ~100 kSa/s in order to record an accurate THD measurement to within 1%.

- A sample rate of 16 samples per cycle produces a good trade-off between sample rate and performance, although of course performance increases as sample rate is increased further. This is because at 15 samples per cycle, the 13th harmonic can contribute, and at 2x BS EN 50160 levels this might have amplitudes of 6%. At 16 samples per cycle, for input frequencies in the range 45-55Hz, the 13th harmonic can no longer alias directly onto the input frequency. Other higher-order harmonics can, but they are much lower magnitude as expected under BS EN 50160.

- The errors due to harmonics cannot be reduced by using the 2nd order algorithms, cancellation techniques, or fixed ½ or 1-cycle post-averaging. The worst errors can, however, be reduced significantly by the use of a novel anti-ripple filter

173

which is introduced later in section 4.3.

- Since this thesis is targeting low sample-rate algorithms in microgrid scenarios with potentially high levels of harmonics, any critical algorithms should key off the Fourier fundamental measurements, since these remain accurate. The all-harmonic RMS and THD measurements should be used only for indication purposes.

4.2.8   Effect of increasing the number of base measurement cycles

Increasing the number of base Fourier/RMS measurement cycles does not decrease the magnitude of the largest errors due to aliased harmonic contamination – it simply narrows the frequency windows over which those largest errors occur. Referring back to Figure 4-4 & Figure 4-5, repeating the analysis (which excludes the effect of anti-aliasing filters, ADC and DC block) but with the measurements taken over exactly 5 cycles results in the following two plots:
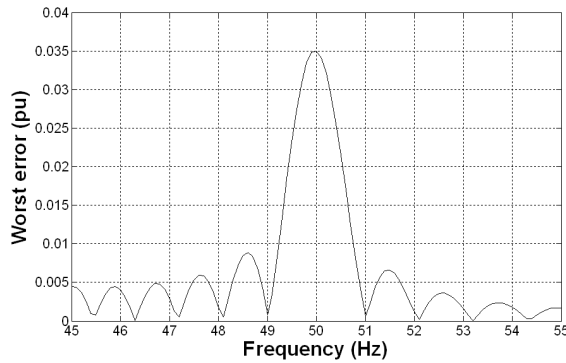
Figure 4-24 : Prediction of errors on the Fourier fundamental measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz, using 5 cycles
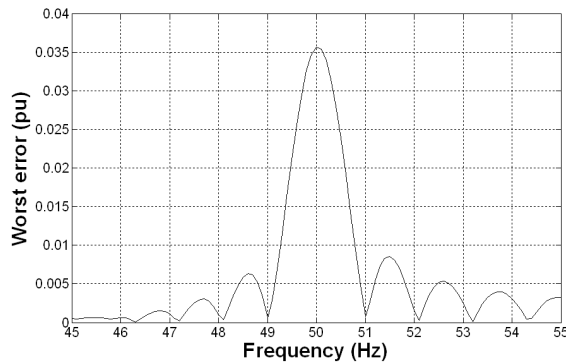
Figure 4-25 : Prediction of errors on the all-harmonic RMS measurement at 500Sa/s due to 11th harmonic @ 3.5%, for input frequencies in the range 45-55Hz, using 5 cycles

It can be seen that the peak error is still 0.035pu, but the error occurs in a tighter

frequency band due to the longer sampling window. Because the frequency windows at which the peak errors appear are smaller, there is less chance of different harmonics causing coincident peak errors through aliasing. However, the overall performance is dominated by the error from the worst aliased harmonic, so this effect does not help to reduce the potential peak error. This has been verified by re-running the simulations of section 4.2.6 but with 5 cycles of Fourier and RMS integration. The overall worst errors are not reduced significantly.

4.2.8.1   Findings for this section

- Increasing the number of measurement cycles does not help reduce the magnitude of the errors; it simply tightens the windows of input frequency over which the worst errors occur.

4.3   Addition of a novel ripple-removal filter to minimise aliased harmonic effects

Thus far in chapters 3 & 4, there are several key findings which have been made. These findings suggest that some of the targets of chapter 2 are likely to be met with ease, but that other targets are difficult to meet. In terms of voltage measurement accuracy, the worst case ripple errors at the outputs of a 1 or 1½-cycle measurement at a sample rate of 500 Sa/s (10 samples per cycle) are approximately ±0.02pu due to aliased harmonics, anti-aliasing filters, noise, ADC quantisation and processing (see Figure 4-20). This is sufficient to meet the target for relaying operations give in Table 2-11, but much too large to meet the desired ±0.001pu specification of Table 2-12 for control applications. Table 2-12 does, however, allow up to 5 cycles for a measurement used for control purposes, in order to allow further attenuation of such errors. This additional time available for the measurement can be used advantageously.

Referring back to Figure 4-4 and Figure 4-6, it has been found that the largest errors due to aliased harmonics tend to occur as sub-harmonic ripple on the measurements. These errors are also larger than any errors due to integration/interpolation, noise, ADC quantisation, and un-aliased harmonics. The worst case (largest ripple magnitude) is when the ripple frequency falls very close to (but not exactly at) 0 Hz. In this case, only very long (more than 5 periods) stages of post-processing can remove the error. A special case is where the ripple frequency falls at 0 Hz exactly. In this case, there is no ripple on the measurement output, but an fixed absolute error is incurred. Intermediate frequency ripples, however, can be removed entirely by a further stage of exact-time averaging over a time of $1/f_R$ seconds, where the ripple occurs at f_R Hz.

The problem in the case of aliased harmonics is that the ripple can occur at any "random" frequency, dependent upon the input signal harmonic content. The maximum time length of the FIR filter is also limited by the maximum latency which we require of the measurement. For control purposes, according to Table 2-12, the measurement time latency requirement is 5 cycles (100ms). This allows for the 1½ cycle measurement blocks proposed thus far, plus an additional 3½ cycles of post-processing. This means that, for a 50 Hz system, a final post-averaging filter of maximum time length 3.5/50=70ms could be applied. This filter could, in theory, be used to entirely reject ripple down to 14.3 Hz. Allowing longer latencies would obviously allow even lower frequency ripples to be rejected, with further benefit. Remember, however, that it will never be possible to remove the DC errors at 0 Hz ripple.

4.3.1   Design of a novel, adaptive, ripple-rejection filter

Such a novel filter to reject ripple at unknown frequencies, using the FIR exact-time averaging blocks from section 3.2, has been developed during the course of this thesis. Its design is shown below. This filters provides additional noise rejection as well as ripple rejection.
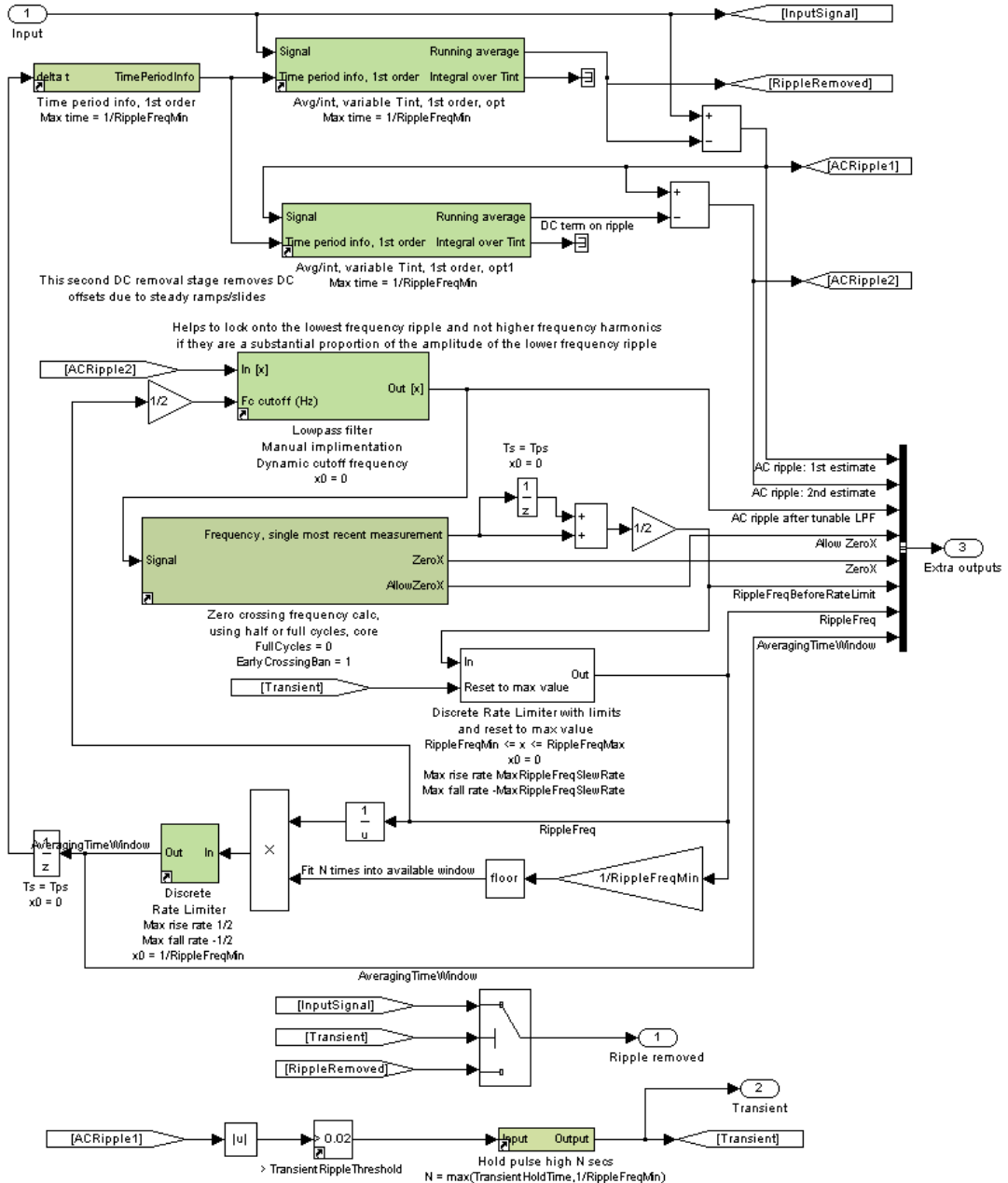
Figure 4-26 : Design of a novel, adaptive FIR ripple-removal filter

The design of the algorithm shown in Figure 4-26, at the core, uses a single exact-time averaging block to filter the signal. The maximum time length of this filter (and thus its maximum contribution to measurement latency) is set by the reciprocal of the parameter *RippleFreqMin*, which is set at compile-time. The trick inside this filter is the determination of the actual time length to average over. The processes can be summarised as follows:

1. Perform the exact-time averaging, using the determined time length (fed back from step 8).

2. The input signal minus the averaged signal reveals the 1st estimate of the AC ripple. This 1st estimate of AC ripple remains at a non-zero level during steady ramps of the input signal, which is undesirable. Therefore, this signal is averaged again over the same timeframe, and a second subtraction is performed to give a 2nd estimate of AC ripple. This 2nd estimate is zero for a steady linear ramp on the input signal.

3. Low-pass filter the AC ripple signal through a single, 1st-order low-pass filter with a dynamically adjustable cut-off frequency. This cut-off frequency is set to ½ the current estimate of ripple frequency (fed back from step 8). This reduces noise and attenuates ripple at higher frequencies. The idea is to try and pick the lowest ripple frequency from the signal, without carrying out a full FFT in real-time (which would require a large amount of CPU time).

4. Measure the frequency of the low-pass filtered AC ripple signal, using zero crossings. The detail of the zero crossings algorithm to do this is shown in Fig. E-8.

5. Slew-rate limit the measured ripple frequency, and limit the frequency within the bounds $RippleFreqMin \leq F \leq RippleFreqMax$

6. The smallest time average window required to reject the ripple frequency f would be $1/f$. However, instead, when the ripple frequency f is large, this might result in a small averaging window, which is not ideal for noise rejection. Therefore, a further step calculates the number of integer windows of length $1/f$ that can be fitted inside the allowed time window $1/RippleFreqMin$. The desired averaging time window is thus $1/f*\text{floor}(f/RippleFreqMin)$.

7. The result from step 6 above is that the averaging time window may jump suddenly when the floor() function crosses an integer threshold. This can cause high-bandwidth steps at the output (sudden jumps), which may be undesirable. Thus, a second slew rate filter is inserted. The maximum rate for this is calculated as a maximum time window change of $0.5/RippleFreqMin$ (the biggest step we would expect to make) divided by the time required to settle. This is $1/RippleFreqMin$, the maximum time length of the averaging stage. Thus the slew rate turns out to be simply ½.

8. Feed back the result to the rest of the algorithm, with the required 1-sample state delay

The exceptions to this process occur when a transient is detected in the input waveform. This is determined by a threshold set on the magnitude of the AC signal content. When this

occurs, the following process over-ride occurs:-

- The first slew-rate filter is reset to the parameter *RippleFreqMax*. This tends to set the averaging time window up towards the maximum allowed length (as many integer ripple periods can be fit into the maximum time window).

- The input is fed directly to the output, bypassing the averaging stage entirely. This happens for at least 1/*RippleFreqMin* seconds, the maximum time taken for the averaging filter to settle to the new input value. The filter thus has zero latency during the transient events. This is highly desirable!

- Optionally, the reset/bypass state can be held for an additional pre-determined time period. This extends the amount of time following transient detection, for which the slew rate filter is forced to hold the value *RippleFreqMax*, and the input is fed directly to the output.

To test this block, a simple Simulink simulation is used. This simulation applies the following "genuine" signal:

- A signal of amplitude 1, with a dip from 1.0 to 0.8 between 7 & 7.03 seconds, and a rise from 0.8 to 1 again between 7.1 & 7.13 seconds. This simulates the way a voltage dip would be measured by a 1½-cycle Fourier measurement block.

To this "genuine" signal, two errors are added, both of 0.01pu peak magnitude. One is at a fixed ripple frequency of 75Hz. The other is at a variable frequency which starts at 100Hz, ramps down to 0Hz at t=5s, and then back up to 100Hz at t=10s, the end of the simulation.

The ripple removal filter is set with the following parameters:

- *RippleFreqMin* = 50/3.5 (14.3 Hz), a maximum time length of 3.5 cycles at 50Hz, or 70ms

- *RippleFreqMax* = 100

- *MaxRippleSlewRate* = 500

- *TransientRippleThreshold* = 0.05

- *TransientHoldTime* = 0 (clipped up to 1/*RippleMinFreq* = 70ms inside the algorithm)

The ripple remover succeeds in removing most of the ripple due to both the interfering ripple signals, as can be seen in Figure 4-27 to Figure 4-29. The main points to note are:-

- At *t*=7s and *t*=7.1s, the filter detects the transient. For 70ms after each of these

points the signal passes straight through the filter with zero latency. In this simulation, the transient events merge together as the fall and rise are close together.

- At other times, the ripple is mostly removed.

- An exception is the first 70ms of the simulation, when essentially a transient is detected due to the sudden application of the waveform.

- Another exception is at about t=5s (see Figure 4-28 & Figure 4-29). During this time, the frequency of one of the rippling error input waveforms is close to 0Hz, DC. This means that the averaging filter cannot remove it within the allowed timeframe.

- The algorithm is correctly able to lock on to the lower frequency of the two rippling error waveforms, even though they are at the same amplitude. The filter tracks the variable rate ripple error when it drops below 75Hz, and it tracks the 75Hz ripple error when the variable rate ripple error frequency rises above 75Hz.
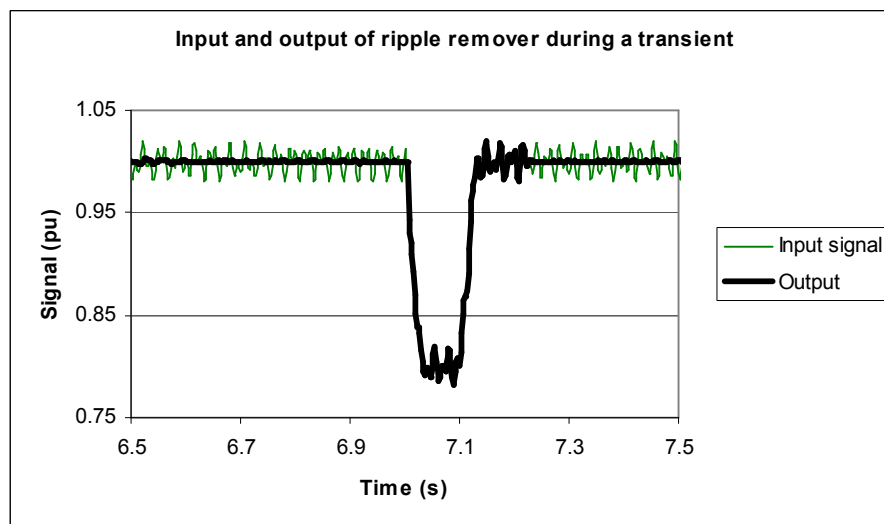
Figure 4-27 : Input and output of a ripple removal filter during a transient
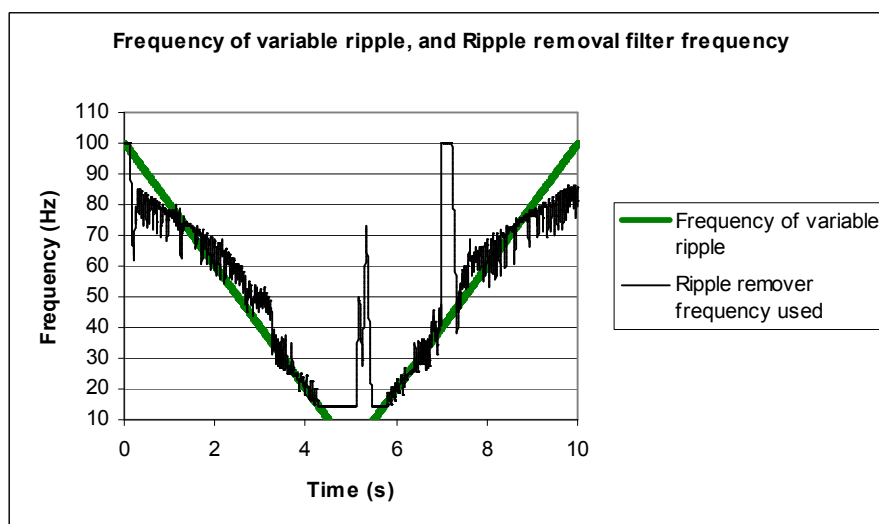
Figure 4-28 : Frequency of variable ripple vs. actual Ripple removal frequency
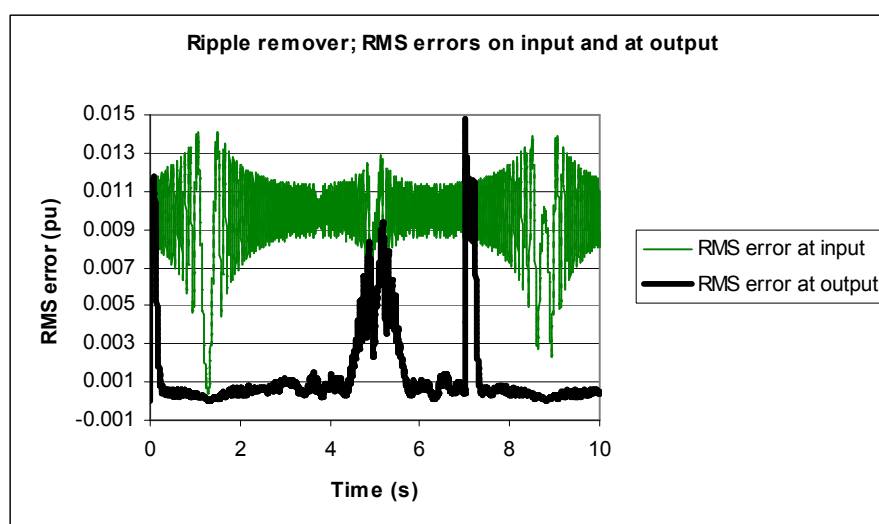
Figure 4-29 : Demonstration of ripple attenuation using the ripple removal filter

4.3.2 Assessment of errors on the fundamental amplitude and phase measurement using a 5-cycle latency measurement and worst-case microgrid harmonics at 500Sa/s

To create a lowest-ripple 5-cycle latency measurement of fundamental voltage or current amplitude, which can be used for delicate control purposes within environments of high harmonic content and noise, at low sample rates, the proposed method is therefore:-

- use a 1½ cycle measurement algorithm made up of a base Fourier stage over 1 cycle (without 2nd harmonic cancellation), followed by an additional ½-cycle averaging …

- followed by an adaptive ripple-removal filter, of maximum latency 3½ cycles. This means that the lowest error ripple frequency due to aliased harmonics which can

181

be removed fully is at 14.3Hz. This cannot remove all ripple errors, but it can remove some errors almost totally, and it will attenuate all non-DC ripple errors by some amount. Since the ripple errors due to aliased harmonics can in the worst case scenarios be considered to act additively rather than in an RMS fashion (see section 4.2.5.1), any error ripple removal may have a large effect on the final answer, even if not all ripple (due to all aliased harmonics) can be removed.

To assess the likely performance of the ripple removal filter in a real scenario, the simulation of section 4.2.6, using worst-case microgrid harmonics, can be repeated, but this time with the ripple-removal filter added.
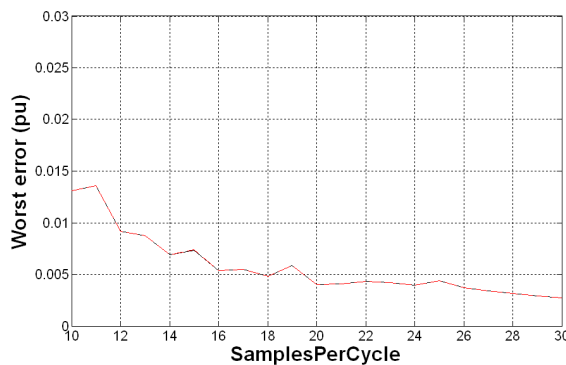
Figure 4-30 : Simulation of worst overall errors on Fourier fundamental measurements, due to worst case microgrid harmonics 2-40 applied all at once, plus noise and ADC quantisation, for 1½ cycle measurement plus ripple removal filter. 1st order (solid) and 2nd order (red dashes) methods

The beneficial effect of adding the ripple-removal filter can be seen by comparison of Figure 4-30 with Figure 4-20. The worst errors at 10 Sa/cycle are reduced from 0.021 to 0.013pu.

4.4   Summary of key findings without ADC oversampling

Methods for Fourier and all-harmonic amplitude have been presented in, which incorporate a number of novel features based on robust FIR filters:-

- Adaptions of an existing SimPowerSystems block which provide more robust and accurate methods of evaluating exact-time average values.

- The application of these filters in Fourier measurement stages with extra novel cascaded averaging steps to create measurements with a latency of 1½ cycles. This almost entirely eliminates integration/interpolation ripple and the effects of errors on the frequency measurement, even at sample rates as low as 10 samples per cycle.

- The option of tapping off lower-latency measurements with ½-cycle and 1-cycle

timeframe is included in the blocks. This includes the option of a 1-cycle latency output with lower integration/interpolation error by using a 2nd-harmonic cancellation technique adapted from a published PLL.

- A selection process has been designed to choose the appropriate combination of measurement blocks to use in a given scenario (section 3.13.1).

- The option of extending the post-processing with a novel, adaptive ripple-removal filter has been presented. This creates a total measurement system consisting of 3 cascaded FIR averaging stages of lengths 1-cycle, ½-cycle and approximately 3½ cycles, each implemented in a specific way for a specific purpose. The total measurement latency (within the digital domain) is approximately 5 cycles.

- An effective set of algorithms has been created to measure fundamental amplitude and phase. Their performance tabulated below is limited due to the extremely strict constraints being applied in this thesis: low sample rates, high noise, and high harmonic contamination. In scenarios which allow higher sample rates, lower noise, or lower THD, the identical algorithms will give much more accurate results.

Sample rates down to 10 samples per cycle (500 Sa/s) have been shown to be usable, so long as any controls or relays key off the fundamental amplitude measurement, and not the all-harmonic RMS measurement. The all-harmonic RMS and THD measurements should be used for indication only, unless the sample rate used is much higher (and the low-pass filter cutoff frequency is raised significantly).

The worst case errors in a microgrid scenario are dominated by the effects of aliased harmonics. The magnitude of the worst case errors including all effects such as ADC quantisation, un-aliased harmonics, aliased harmonics, integration/interpolation error, and frequency measurement error, for a 10 sample-per-cycle system at nominally 50Hz in the presence of 53% THD_V harmonics are approximately:

	Fundamental amplitude error (peak pu)	Fundamental amplitude ripple magnitude (±pu)	Fundamental phase error (peak degrees)
0.5-cycle latency measurement "0.5(NC)"	0.042	0.042	3.2
1-cycle latency measurement "1(NC)"	0.021	0.019	1.4
1-cycle (with 2nd harmonic cancellation) "1(C)"	0.021	0.020	1.5
1.5-cycle latency measurement "1(NC)+0.5"	0.018	0.017	1.2
5-cycle latency measurement "1(NC)+0.5+RippleRemoval"	0.013	0.013	0.8

Table 4-2 : Worst case Fourier fundamental measurement errors for voltage waveforms with up to 53% THD_V, at 500Sa/s

In Table 4-2, the conditions are:-

- Harmonics applied as per Table 2-5, THD$_V$=53%

- Anti-alias filter using a 2nd order low-pass filter (2 cascaded RC filters), f_c=125Hz

- Instrumentation noise (Gaussian), post-filtering, 0.005pu RMS (46dB SNR)

- ADC scaling over -2 to +2 pu, with 12 bits, with 2 bits RMS ADC noise (over and above normal quantisation noise)

- ADC sampling and processing at 500 Sa/s (nominally 10 Sa/cycle @ 50Hz)

As described above and in section 2.7.2, this is a very extreme test, and any microgrid operating with such voltage harmonic content for any length of time would suffer from severe problems such as transformer heating/loss etc.

It is also important to stress that these are the worst absolute (instantaneous) errors found for any input frequency in the range 45 to 55 Hz. Thus, normally, even for the worst input frequency, the RMS error is less than this figure by at least a factor of $\sqrt{2}$ and usually by a greater factor. When the data is examined across the range of input frequencies, it is also possible to see that the largest errors only occur at certain input frequencies. These frequencies are the ones at which harmonics alias onto perfectly onto the fundamental, to produce an almost DC error term which cannot be removed by the ripple-removal filters. This shows up on Figure 4-31 and Figure 4-32 below. Figure 4-31 shows the amplitude errors from the "1(NC)+0.5" and "1(NC)+0.5+Ripple removal" measurement systems. The worst input frequency in this case is around 50Hz, where, for example, the 11th harmonic can alias directly onto the fundamental, as $f_{Nyquist}$ is 250 Hz. At other input frequencies, the ripple removal filter can successfully attenuate the errors by much larger amounts.

An additional point of note is that during the above experiment, the threshold for transient detection within the ripple filter can be set as low as about 0.02pu without causing any spurious transient detections. This means that the ripple-removal filter can be applied to the 1½-cycle measurement, and any transient of > ±0.02pu magnitude will be tracked with only the 1½-cycle latency, as the ripple removal filter will pass the data straight through during the transient. During steady-state operation, the ripple-removal filter will automatically switch on and provide an output with much lower ripple, suitable for control purposes.
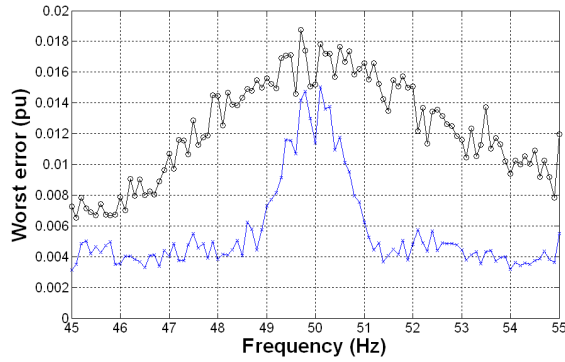
Figure 4-31 : Worst case Fourier fundamental amplitude measurement errors (pu) for voltage waveforms with up to 53% THD$_V$, at 500Sa/s, against input frequency. Without (black o) and with (blue x) ripple removal filter
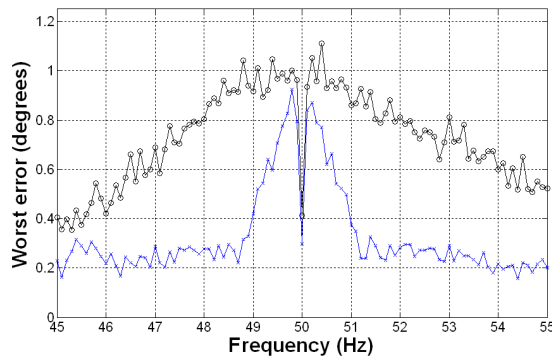
Figure 4-32 : Worst case Fourier fundamental phase measurement errors (degrees) for voltage waveforms with up to 53% THD$_V$, at 500Sa/s, against input frequency. Without (black o) and with (blue x) ripple removal filter

When the phase error is examined against frequency, a slightly different phenomena is observed. In this case the errors are again worst around 50Hz, but at exactly 50Hz the error appears to drop sharply. This is because, in the applied waveform, the odd harmonics are all applied at a coherent zero phase relative to the fundamental, which gives the worst case amplitude errors (see sections 2.7.2 and 4.2.2). If these phases were randomised a little, then Figure 4-32 would not show the dip at exactly 50Hz, and the figure should be interpreted as if that were the case.

The analysis can easily be repeated with a scenario which is relaxed to the harmonic content listed in

Table 2-6, (THD$_V$=28.2%). This waveform still contains approximately double the THD which would arise if all harmonics 2-40 were added together at the maximum individual levels allowed under BS EN 50160. This exceeds the total BS EN 50160 specification for total THD (8%) by a factor of 3.5, and is still a severe test scenario. Under these conditions, the

table of worst expected errors reduces to:

	Fundamental amplitude absolute error (peak pu)	Fundamental amplitude ripple magnitude (±pu)	Fundamental phase error (peak degrees)
0.5-cycle latency measurement "0.5(NC)"	0.039	0.035	2.5
1-cycle latency measurement "1(NC)"	0.014	0.013	1.0
1-cycle (with 2nd harmonic cancellation) "1(C)"	0.017	0.014	1.0
1.5-cycle latency measurement "1(NC)+0.5"	0.012	0.012	0.8
5-cycle latency measurement "1(NC)+0.5+RippleRemoval"	0.009	0.009	0.6

Table 4-3 : Worst case Fourier fundamental measurement errors for voltage waveforms with up to 28.2% THD$_V$, at 500Sa/s

The data from Table 4-2 and Table 4-3 should not be used to predict approximate errors for THD levels below 28.2% via extrapolation based upon a lower known THD level. This is because below this level, the noise and ADC noise/quantisation errors may become dominant (see Figure 3-70 and Figure 3-71). Increasing the sample rate or decreasing the instrumentation noise can both reduce these errors. Increasing the sample rate initially will cause the errors to vary approximately with SampleRate^{-1}, as shown by Figure 4-21. This rate is a combination of the decrease in noise error which varies with SampleRate$^{-\frac{1}{2}}$ and the decrease in errors due to aliased harmonics, integration/interpolation etc which falls faster than SampleRate^{-1}. The errors due to aliased harmonics drop almost completely to zero when sample rate is above 80 samples per cycle, this is because above this level all the harmonics 2-40 are not aliased. Only the harmonics above 40 will still cause problems.

Below, a third table shows how the error magnitudes change for the 28.2% THD$_V$ case, if the instrumentation is improved:

- Instrumentation noise is lowered from 0.005pu to 0.001pu RMS (46dB to 60dB SNR)
- ADC noise (above normal quantisation) is reduced from 2 bits RMS to 1 bit RMS
- Sample rate is increased from 10 Sa/cycle to 16 Sa/cycle @ 50 Hz (800 Sa/s)

	Fundamental amplitude error (peak pu)	Fundamental amplitude ripple magnitude (±pu)	Fundamental phase error (peak degrees)
0.5-cycle latency measurement "0.5(NC)"	0.028	0.028	1.5
1-cycle latency measurement "1(NC)"	0.004	0.004	0.25
1-cycle (with 2nd harmonic cancellation) "1(C)"	0.004	0.004	0.25
1.5-cycle latency measurement "1(NC)+0.5"	0.003	0.003	0.25
5-cycle latency measurement "1(NC)+0.5+RippleRemoval"	0.002	0.002	0.2

Table 4-4 : Worst case Fourier fundamental measurement errors for voltage waveforms with up to 28.2% THD$_V$, at 800Sa/s

Clearly, comparing Table 4-3 with Table 4-4, the decrease in noise, and particularly the increase in sample rate to 16 Sa/cycle, reduce the worst case errors and ripple substantially. This was predicted in section 4.2.7

Also, it can be seen that the ripple magnitudes are almost identical to the peak absolute error magnitudes. This is because the worst errors usually occur due to low-frequency (sub-14Hz) ripples. These don't cause any DC offset to the overall measurement, hence the absolute error peak magnitude is almost identical to the 1-sided ripple magnitude.

An important conclusion at this point is that the outputs of the 1½-cycle and 5-cycle measurements meet all but one of the requirements laid down in section 2.10. The amplitude ripple target of ±0.001pu with a 5-cycle measurement for control purposes (Table 2-12) cannot be met by a digital measurement using sampling at below 17 samples per cycle for the worst microgrid harmonic content cases with 53% or 28% THD_V. This is due to low-frequency ripples on the Fourier amplitude measurements caused by aliased harmonics. Ripple at >14Hz can be removed by the ripple removal filter, but when the ripple is at lower frequencies than this, approaching DC, the ripple cannot be removed by the post-filtering within the allowable 5-cycle measurement latency time. There are 3 possible solutions to this problem:-

1) Increase the sample rate of the entire system above 16 samples per cycle. This may not be possible, due to the overall CPU burden.

2) Increase the performance of the analogue-anti-aliasing filters. Designing such an analogue filter with increased attenuation above the cut-off frequency, without increasing the group delay significantly, is an extremely difficult or impossible task. Such a filter would need to be individually designed for any specified cut-off frequency, taking into account circuit layout and parasitic component tolerances which may change between component batches (Kuhlmann, 2007).

3) A third solution is to clock the ADC at a higher rate than 16 samples per cycle and perform a very simple digital pre-processing/filtering task at this higher sample rate to remove the harmonics which would otherwise cause the worst errors. Then the data can be down-sampled and passed to the above algorithms at sample rates down to 10 samples per cycle. This solution is the chosen method, and is examined in further in section 4.5.

A further summary revision of findings thus far is that:-

- The sample rate does not need to be an integer multiple of the nominal frequency and synchronisation with zero-crossings is not required.

- For simulation purposes, the 2nd order algorithms are useful for reducing the ripple on the measurements due to integration and interpolation.

- In real applications with noise and harmonic content, the advantage of the 2nd order algorithms is usually lost. The dominant effects are noise content, ADC quantisation, and aliased harmonics above the Nyquist frequency. The 2nd order algorithms can still safely be used since they do not produce any worse results than the 1st order algorithms.

- The 1-cycle latency algorithm using 2nd harmonic cancellation can produce worse results than the 1-cycle latency algorithm without cancellation, under conditions of high noise and THD (see Table 4-3). This is due to oscillations which may result due to the FIR nature of the cancellation algorithm. Therefore, unless instrumentation noise and harmonic content is low, this technique is best avoided. It also adds significantly to the burden on the CPU.

- The all-harmonic RMS measurements at 10 samples per cycle can be inaccurate by up to 15% in the worst-case microgrid scenario described, with a tendency to read low. The error cannot be removed by post-processing averaging.

- THD measurement at 10 sample per cycle can be in error by as much as 50% in the scenario described (normally reading low), and the error cannot be removed by post-processing averaging.

4.5   Addition of ADC over-sampling and notch pre-filtering stage before down-sampling

In the previous section, it was concluded that the only available way to achieve the amplitude measurement ripple target (for control purposes) of ±0.001pu (from Table 2-12), while clocking the bulk of the algorithm at 16 samples per cycle or slower, is to apply some simple ADC oversampling and pre-filtering to remove the worst harmonics, before down-sampling to the normal clock rate in the main algorithms.

A repeat of the analysis of section 4.2.5.2 (which is a prediction of the magnitude of the expected errors, and does not account for the ripple removal filter of section 4.3), allows us to predict the worst offending aliased harmonics for any proposed sample rate of the main algorithms. Taking a target of 10 samples per cycle with a 50 Hz nominal frequency, this is a sample rate of 500 Sa/s. The analysis under this scenario shows the magnitudes of the errors (ripples) due to different harmonics becoming aliased when sampled at this rate. The results are shown below in Figure 4-33.
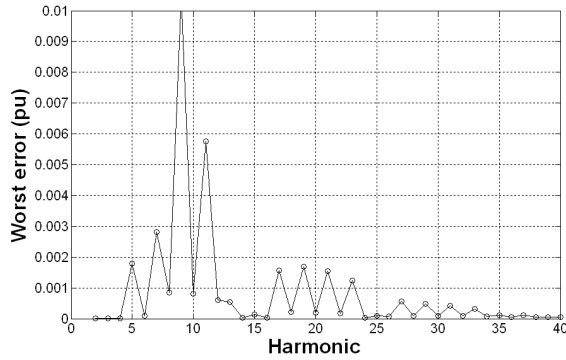
Figure 4-33 : Relative effect of aliased harmonics on errors/ripple, ignoring the effect of the ripple-removal filter

The target is to get below ±0.001pu ripple, therefore it may be necessary to remove all the following harmonics: 5th,.7th, 9th, 11th, 17th, 19th, 21st & 23rd.

However, further analysis of the potential ripple frequencies due to the aliased harmonics, for any fundamental in the range 45 to 55Hz, shows that the ripples due to some of these harmonics can be removed using the ripple removal filter. Only those that can result in ripples at frequencies of less than ≈15Hz will pass un-attenuated through the ripple removal filter.

Harmonic	Harmonic frequency range for fundamentals in the range 45-50-55Hz	Expected maximum ripple error magnitude due to digital processing at 500Sa/s (10Sa/cycle @ 50Hz), For any fundamental in the range 45-55Hz	Lowest ripple frequency after Fourier analysis, due to digital processing at 500Sa/s (10Sa/cycle @ 50Hz), For any fundamental in the range 45-55Hz. Also, harmonic frequency "danger" range to cause <15Hz ripple.	Needs to be removed ?
5th	225-250-275 Hz	0.002	170	NO
7th	315-350-385 Hz	0.003	60	NO
9th	405-450-495 Hz	0.01	0 (436.5 to 463.5 Hz danger area)	YES, by 20dB in danger area
11th	495-550-605 Hz	0.006	0 (533.5 to 566.5 Hz danger area)	YES, by 15dB in danger area
17th	765-850-935 Hz	0.0015	15	YES, by 3.5dB
19th	855-950-1045 Hz	0.0015	0	YES, by 3.5dB
21st	945-1050-1155 Hz	0.0015	0	YES, by 3.5dB
23rd	1035-1150-1265 Hz	0.0012	0	YES, by 1.6dB

Table 4-5 : Harmonics to attenuate using over-sampling and pre-filtering

So, there are 6 harmonics which, if removed from the final signal at 500 Sa/s, should allow the ±0.001pu ripple specification to be approached. This needs an initial ADC sampling of at least 60 Sa/cycle, 3000 Sa/s, i.e. 6x over-sampling, to be able to successfully capture and filter the 23rd harmonic of a 55 Hz input waveform. The harmonics of greatest significance are the 9th and 11th. (To remove only these two harmonics would require at least 3x over-sampling at 1500 Sa/s).

The next step is to design a filter to reject these harmonics, remembering that each may appear over a range of frequencies for a fundamental in the range 45Hz to 55Hz. The down-sampling factor from a 3000 Sa/s ADC to the 500 Sa/s main processing is 6. This factor sets the number of FIR samples/weightings which are used, and also (after subtracting 1) the number of zeros which can be placed during the FIR filter design.

Note that there are 6 harmonics to be removed, but only 5 available zeros, which must occur in conjugate pairs. Therefore, some of the zeros must be placed strategically to cover multiple harmonics. From Table 4-5, and accounting for the fact that the Nyquist frequency of the 3000 Sa/s ADC system is 1500 Hz, the location of the zeros is chosen to be:-

- 1500Hz (single zero, conjugate of itself)

- ±467Hz (pair of zeros, to cover the lower frequencies of the 9th)

- ±840Hz (pair of zeros, to cover the higher frequencies of the 9th, the 11th, and the higher harmonics)

Such a filter can be designed and optimised using a MATLAB script. It's zeros, poles, and bode plot are shown below. The dashed red line on the bode plot of Figure 4-34 is the specification for required attenuation, defined by Table 4-5. The FIR weightings of the 6 samples are:-

- 0.23850705983587, 0.06142383205831, 0.20006910810582, 0.20006910810582, 0.06142383205831, 0.23850705983587

Note that due to the attenuation at 50Hz, the weightings must also be multiplied by the required value to bring gain at 50Hz back to unity (0dB). Additionally, a polynomial of order 2 can be fitted in the region of 40-50-60Hz to create a fitted function for the gain correction, which covers eventualities when the input signal is not at 50Hz. This process should also be used to correct the phase of the measured waveforms, although the phase correction only requires a linear slope. For the filter designed above, the correction terms

determined by this process are:-

- Gain correction = 0.00000796849357*f^2 - 0.00003983954291*f+1.00072712529167

- Phase correction (radians) = 0.00523598775598*f
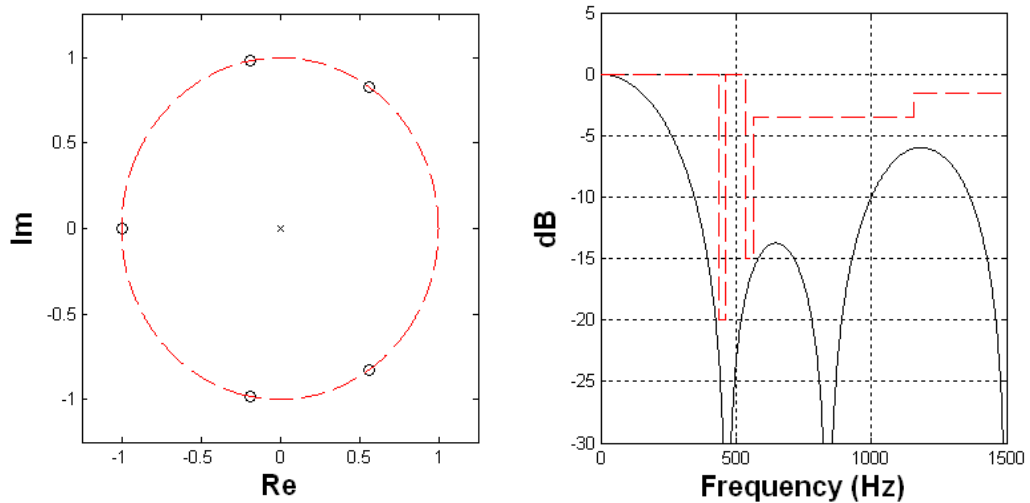
Figure 4-34 : Zeros, poles and bode plot for 3000 Sa/s over-sampling pre-filter (Nyquist frequency 1500 Hz)

Thus, it can be seen that a 6-times over-sampling at the ADC, at 3000 Sa/s, combined with a very simple 6-sample (5-zero) FIR filter, can be used to notch out the worst problem harmonics. After this step, the output of the FIR filter can be down-sampled so that 1 of every 6 results is forwarded to the main processing algorithms at 500 Sa/s. This should allow the main Fourier-based/averaging/ripple removal algorithms at 500 Sa/s to meet or come close to the ±0.001pu ripple specification. Note, the measurements of all-harmonic RMS and THD will definitely be invalid when using this pre-filter, since the 5th and 7th harmonics have been deliberately notched out of any subsequent signal analysis.

Such a process can be carried out on a commercial micro-controller such as the TC1796 by using the peripheral control processor for the ADC and over-sampled algorithms, and then passing the data to the main processor for the lower sample-rate, higher burden algorithms. On the ADI RTS, such a system can be implemented by clocking one of the processor cards, which handles the ADC/DAC inputs/outputs and also carries out the pre-filtering, at the over-sampled rate. The filtered data is then passed via the internal VME bus to one of the other processors at the lower clock frequency (500 Sa/s).

On the ADI RTS at present, the highest theoretical ADC clock rate is 1562.5 Sa/s due to the pas9737 ADC cards which take at least 64*10us=640µs (1/1562.5s) to multiplex though all 64 channels each frame. Therefore, for proving these algorithms on the ADI RTS, the pre-

filtering must be redesigned to allow for only a 1500Sa/s (3x) oversampling, relative to the main 500 Sa/s algorithms. This will not allow the overall measurement system to meet the ±0.001pu ripple specification for 53% THD waveforms, but for cleaner waveforms it will give the best results possible with the current hardware constraints. Since THD on the Strathclyde microgrid at present is normally better than 5%, this does not present a problem in the short term.

For the 1500 Sa/s (3x over-sampling) version, the zeros are placed at ±482Hz. The filter weights are [0.34883448605115, 0.30233102789770, 0.34883448605115]
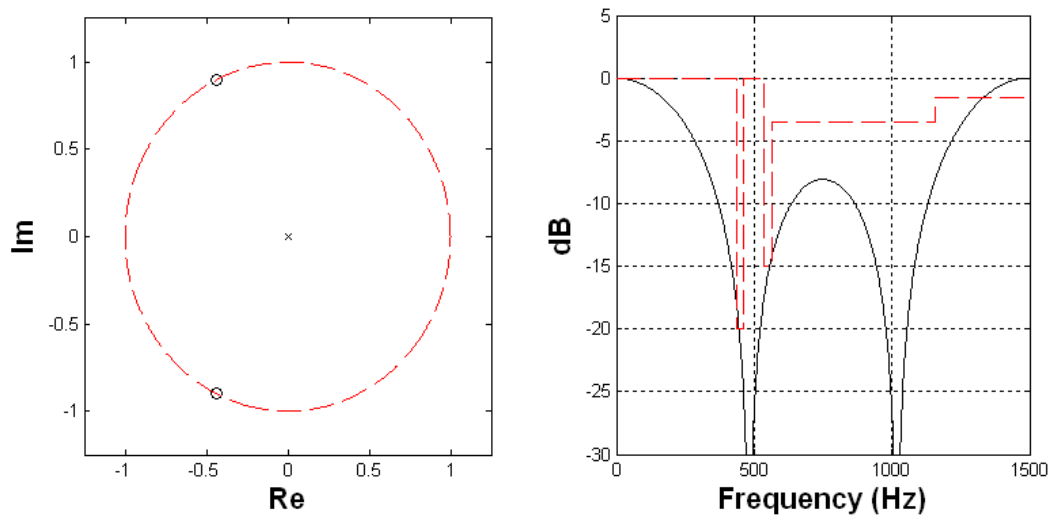
Figure 4-35 : Zeros, poles and bode plot for 1500 Sa/s over-sampling pre-filter (Nyquist frequency 750 Hz)

4.5.1   Analysis of performance improvement using 3000Sa/s over-sampled ADC and FIR pre-filtering

ADC over-sampling and FIR pre-filtering at 3000 Sa/s was added to the simulations of section 4.4. This results in the following table of errors for the worst microgrid case. This table is generated using the identical test conditions to those used for Table 4-2.

	Fundamental amplitude error (peak pu)	Fundamental amplitude ripple magnitude (±pu)	Fundamental phase error (peak degrees)
0.5-cycle latency measurement "0.5(NC)"	0.027	0.025	1.8
1-cycle latency measurement "1(NC)"	0.0100	0.0075	0.55
1-cycle (with 2nd harmonic cancellation) "1(C)"	0.0080	0.0072	0.52
1.5-cycle latency measurement "1(NC)+0.5"	0.0045	0.0040	0.27
5-cycle latency measurement "1(NC)+0.5+RippleRemoval"	0.0028	0.0023	0.17

Table 4-6 : Worst case Fourier fundamental measurement errors for voltage waveforms with up to 53% THD$_V$, at 500Sa/s, using 6x over-sampled FIR pre-filter

An additional point of note is that during the above experiment, the threshold for transient detection within the ripple filter can be set as low as about 0.005pu without causing any spurious transient detections. This means that the ripple-removal filter can be applied to the 1½-cycle measurement, and any transient of > ±0.005pu magnitude will be tracked with only the 1½-cycle latency, as the ripple removal filter will pass the data straight through. During steady-state operation, the ripple-removal filter will switch on automatically and provide an output with much lower ripple, suitable for control purposes.

As Table 4-6 shows, the required ripple specification of ±0.001pu is still not quite met at the 5-cycle measurement output. If, however, the instrumentation noise level can be dropped from 0.005pu RMS (46dB SNR) to 0.001pu RMS (60dB SNR), the additional ADC bit noise is dropped from 2 bits RMS to 1 bit RMS, and the THD_V of the waveform is dropped from 53% to 28%, the specification for ±0.001pu ripple on the Fourier amplitude measurement can finally be met. The results are shown in Table 4-7.

	Fundamental amplitude error (peak pu)	Fundamental amplitude ripple magnitude (±pu)	Fundamental phase error (peak degrees)
0.5-cycle latency measurement "0.5(NC)"	0.025	0.023	1.4
1-cycle latency measurement "1(NC)"	0.0051	0.0041	0.27
1-cycle (with 2nd harmonic cancellation) "1(C)"	0.0035	0.0031	0.22
1.5-cycle latency measurement "1(NC)+0.5"	0.0017	0.0014	0.10
5-cycle latency measurement "1(NC)+0.5+RippleRemoval"	0.0011	**0.0010**	0.08

Table 4-7 : Worst case Fourier fundamental measurement errors for voltage waveforms with up to 28.2% THD_V, at 500Sa/s, using 6x over-sampled FIR pre-filter

4.6 Verification of amplitude measurement errors with closed-loop frequency measurement

In chapters 3 & 4, apart from the analysis of 4.1, it was assumed that the measurement of frequency was perfect. In chapter 5, a frequency/amplitude/phase measurement algorithm (the Clarke-FLL hybrid) is designed and presented. In the following section, the amplitude measurement accuracy of this is further tested in a closed-loop situation using suitable test waveforms.

First, test waveform 1B is used (see section 2.11.1). This waveform has 28% THD_V, plus

unbalance and inter-harmonics, plus dynamic frequency and voltage dip events as described in Table 2-16. The frequency profile is shown in Figure 5-18.

The specification lines for the fundamental amplitude measurement on each of the three phases were set at ±0.001pu from the actual applied values (matching the original requirement from Table 2-12), unless frequency was less than 10 Hz in which case a much wider specification was set. Accuracy is not good below 10 Hz, due to the finite length of the correlation/averaging buffers inside the Fourier analysis which are set to ≈50 samples long (i.e. a 10 Hz signal with a sample rate of 500 Sa/s can be measured accurately). From t=8s onwards in the simulations, the magnitudes on each phase are different to each other due to the negative sequence component applied. The specification lines take into account an allowed reaction time of 40ms (nominally 2 cycles at 50Hz, see Table 2-11). This timeframe accounts for the hardware filtering and ADC, plus the 1½ cycle measurement time within the main algorithm (see section 4.7). A ripple-removal filter with maximum latency of 70ms was used (see section 4.3). To meet the reaction time specification of 40ms during transient effects, this filter has to switch itself out automatically and pass the signal straight through.

The cumulative error score for the magnitude measurements, as determined by the same scoring system given in section 5.7 for the frequency tests, is of the order of 0.5 to 0.65 during the waveform 1B test. This means that, on average, over the 60 seconds of the test, the measurement was 50% outside the specification window of ±0.001pu. Closer analysis reveals when and why this occurs.

First, a plot of the phase A voltage magnitude measurement against the specification is shown for the first 16 seconds of the simulation. During this time there are significant amplitude variations due to dips on one, two and three phases.

On all the graphs which follow, the traces are identified as follows:-

- Black solid line : Measurement

- Red dash-dot : Actual synthesised signal amplitude

- Blue dashes : Lower and Upper limit lines
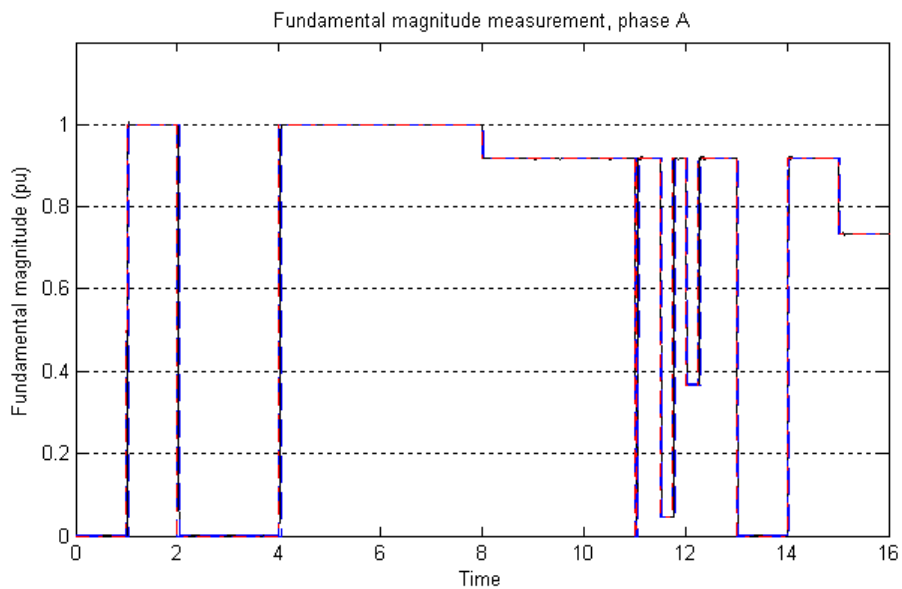
Figure 4-36 : Amplitude measurements, waveform 1B, phase A, t=0 to 16s

Figure 4-36 show the overall shape of the signal magnitude on phase A, and that the measurement follows the signal. The scales do not allow the errors to be examined in detail (no deviations from specification are apparent), and this figure merely gives an overview of the scenario. The plots for the phase B & C measurements look almost identical, and are not shown here. For a more meaningful analysis, the actual measurement errors can be shown at significantly higher magnification for several interesting parts of the scenario. These plots are shown in Figure 4-37 to Figure 4-39.
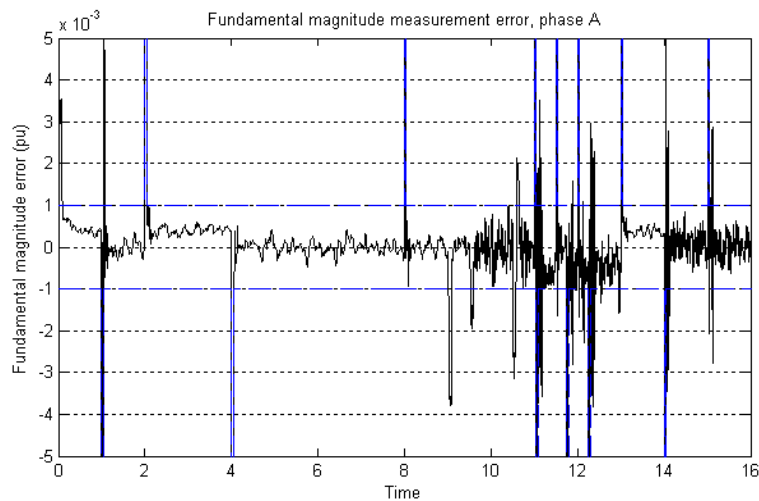
Figure 4-37 : Amplitude measurement errors, waveform 1B, phase A only, t=0 to 16s

The errors, normalised to the actual signal magnitudes, are generally within bounds, save for brief excursions during settling after transients. Even during transients the error is

mostly within specification, since the specifications (blue dashed lines) allow 40ms reaction time, and are significantly wider than ±0.001pu during transients. The plots for phases B & C are almost identical.
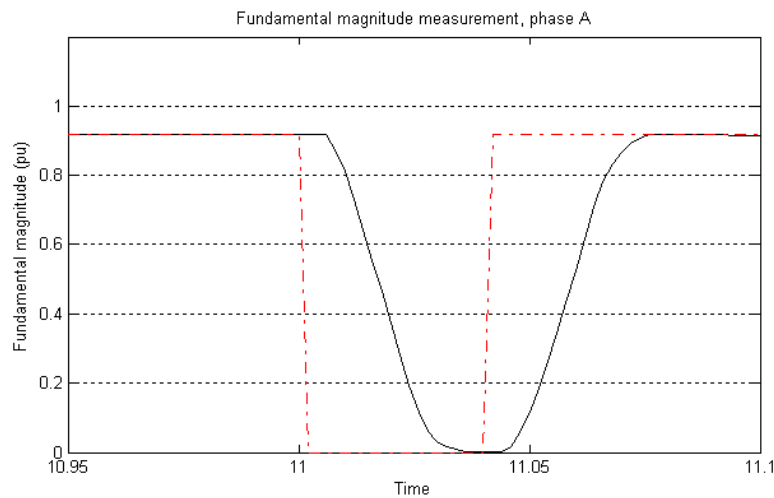
Figure 4-38 : Amplitude measurements, waveform 1B, phase A only, three-phase fault response

To show the measurement latency in detail, the plot above shows the timeframe of the brief three-phase fault (100% dip on all three phases) between t=11s and t=11.04s. The plots for phases B & C are almost identical.
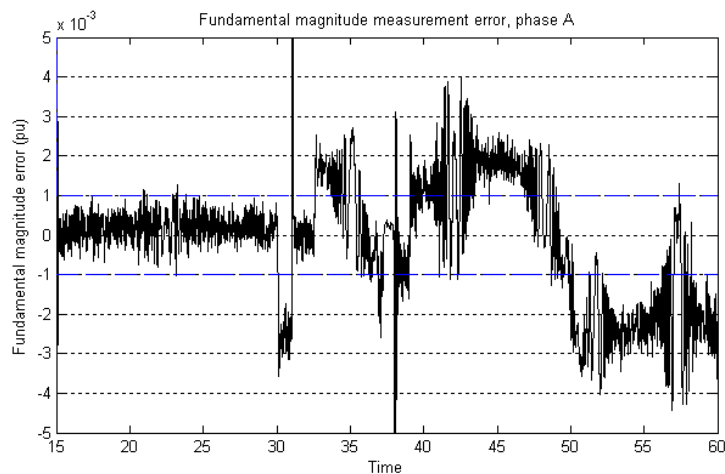
Figure 4-39 : Amplitude measurement errors, waveform 1B, phase A only, t=15 to 60s

The plot above shows the measurement errors over the part of the simulation where the magnitudes are constant, but the frequency changes rapidly (see Figure 5-18). The main point of note is that the absolute specification is violated during the 10 Hz/s frequency ramps, by about a factor of 2. However, much of the ROCOF-induced violation consists of a

DC bias term, which will not cause ripples in any drooped control system. The plots for phases B & C are almost identical.
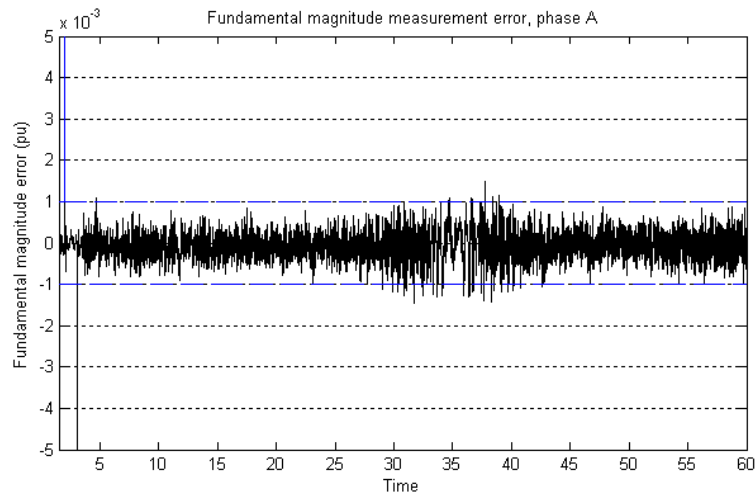
Figure 4-40 : Amplitude measurement errors, waveform 2B, phase A only, t=1.5 to 60s

The plot above shows the measurement errors for waveform 2B (see section 2.11.2), where the signal magnitudes are constant, but the frequency changes slowly (see Figure 4-41). Generally, the specification is met. Slight violations occur during the initial onset of unbalance at t=2s (≈0.003pu) and for certain worst input frequencies in the 49-51Hz bracket (t=30s to t=40s in the simulation, see Figure 4-41), due to the aliased harmonics (mainly the 9th and 11th) which are not fully attenuated within the over-sampled FIR filter (see section 4.5 and Figure 4-31). The plots for phases B & C are almost identical.
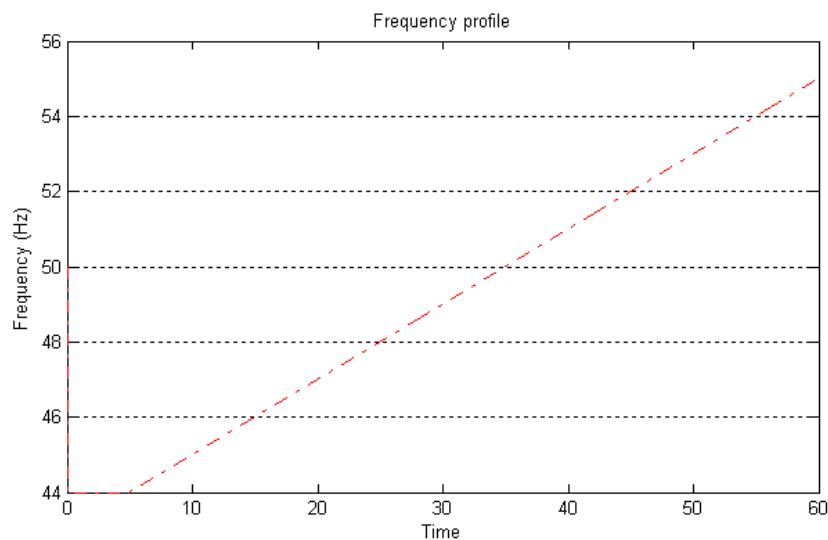
Figure 4-41 : Frequency profile, waveform 2B

The closed-loop frequency measurement does not add any serious problems to either the latency or the ripple of the amplitude measurements. The largest effect is during rapid frequency changes. At a ROCOF of 10 Hz/s, and absolute amplitude error rises to about 0.002pu (see Figure 4-39). However, the error tends to be a constant bias due to the frequency ramp, and is not an additional ripple error. This should not cause an oscillation problem within droop-controlled systems, and is thus tolerable.

The biggest ripple errors still occur, as found earlier, at the "worst frequencies" where aliased harmonics from high-THD waveforms fold back toward the fundamental. For main algorithms operating at 500 Sa/s, with nominal frequencies of 50 Hz, the 9th and 11th harmonic cause the worst problems. This is because for input frequencies near 50 Hz, they arise at about 450 Hz and 550 Hz respectively. These alias upon sampling at 500 Sa/s to almost exactly 50 Hz, and then cause beating with the genuine signal which is close to 50 Hz. The resulting ripple has been significantly reduced from potentially much higher levels by the use of the over-sampled FIR filter of section 4.5, and the novel ripple-removal filter of section 4.3.

Because the ripple-removal filter automatically switches itself out during transients (see sections 4.3 & 4.4 and Figure 4-38), the results shown in this section, which are taken from the output of the ripple-removal filter, generally meet both the low-ripple control requirements (Table 2-12) which require (up to) 5-cycle averaging, and the fast-acting relaying requirements (Table 2-11) which require 1½-cycle digital latency.

4.7   Overall findings from chapters 3 & 4, Amplitude/Phase measurement

Several different measurement architectures have been investigated and compared. The algorithm which is most appropriate depends upon the scenario (which is defined by the AC waveform quality and the measurement/computational hardware constraints). An extremely useful architecture selection process has been created (section 3.13.1). The primary focus of this thesis is upon achieving the best performance in conditions of instrumentation noise and high harmonic content, using the minimum sample rate possible for the bulk of the processing (500 Sa/s target). These constraints have influenced the selection of the optimal measurement architecture.

The architecture is based around cascaded use of FIR filters. FIR filters result in a robust, inherently stable algorithm with a minimum of settling ripple. They have also been shown (see section 3.3) to perform substantially better at noise removal than IIR low-pass

filtering techniques. The cascading of the FIR filters also serves to additionally limit the bandwidth of noise due to the windowing effect caused by the convolution of the filter impulse responses. The overall measurement strategy (with the main algorithms at 500 Sa/s) is shown in Figure 4-42, and includes the following stages:-

- A pair of cascaded RC filters for each phase (implemented by active or passive analogue circuits) with F_c=125Hz (½ Nyquist at 500Sa/s). The filter cut-off frequency cannot be set much lower, or the filter order increased, due to the group delay which is incurred and its effect on transient response.

- ADC 6x oversampling (3000 Sa/s) or 3x oversampling (1500 Sa/s), with the ADC ranged over the -2 to +2 peak voltage span.

- FIR pre-filtering using a simple 6-tap (3kSa/s) or 3-tap (1.5kSa/s) FIR filter to notch out the worst problem harmonics which can alias back onto the fundamental in a 500Sa/s system. The filter notches out the most problematic 9th & 11th harmonics, and also to generally reduce the higher order harmonics.

- Down-sample to 500 Sa/s and pass data to the main processor.

- Removal of DC bias due to instrumentation, using a novel DC block based upon FIR averaging filters (section 3.4.4.2) which can be used over arbitrary timeframes, not necessarily integer multiples of the sample time. This filter has a zero group delay for the AC signals.

- Correction for gain slopes in the analogue low-pass filter and FIR filter, ADC channel-channel timing skews, and calculation of phase corrections which can be applied later in the process.

- Fourier correlations over a single or a half-cycle, optimised for speed, robustness, and re-use of calculations. Again, this uses the FIR averaging filter over an arbitrary timeframe.

- Additional averaging over a half or single cycle (using the FIR averaging filter again). This creates a 1½ cycle latency measurement which has virtually zero integration/interpolation error despite the low sample rate used (500Sa/s) (see section 3.9).

- Further averaging using a novel, adaptive ripple-removal filter which adjusts its latency from 0 (during transients >±0.005pu) to 3.5 cycles (70ms) so as to remove as much of the remaining ripple and noise as possible (see section 4.3). This algorithm again uses the FIR averaging filter which can average over an exact timeframe.

- Sequence and power flow analysis

- Final absolute phase calibrations
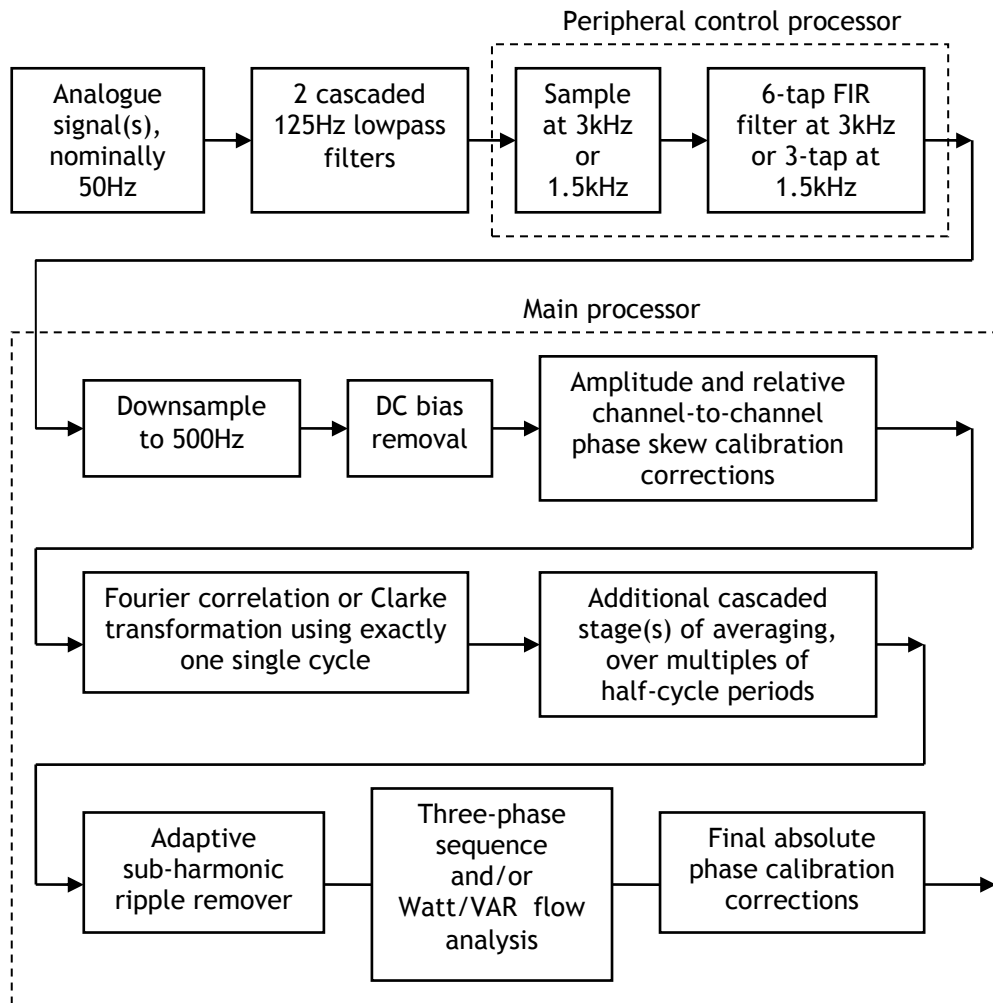
Figure 4-42 : Summary of the optimal measurement configuration for amplitude and phase measurement.

The final output is a ≈5½-cycle latency measurement (≈2-cycle during transients) of extremely high precision and low ripple. The absolute error is limited mainly by the accuracy to which the peripheral hardware (VTs, isolation amplifiers, low-pass filters etc.) can be calibrated. The ripple error is dominated by the harmonic and noise content which remains after all the processing. The actual measurement latency is slightly longer than the 5 cycles taken by the main digital algorithms. This is due to an additional 1/6th cycle (steady state) to ½ cycle (transient) latency due to the analogue anti-aliasing filter section which is a pair of cascaded low-pass filters with cut-off set to f_c=125 Hz as described in section 3.4.5. A further latency of 5/12[ths] of a sample[1] (at 500 Sa/s) (1/24th of a cycle)

---

[1] The delay is calculated by (6-1)/2/6 where 6 is the over-sampling rate. This is the time from the latest sample back to the middle of the last set of over-samples leading up to the current sample.

delay is incurred in the ADC/pre-filtering section which executes at 3000 Sa/s (6x over-sampling).

The total measurement latency for the 5-cycle measurement will thus be approximately 5 + ½ + 1/24 = 5.54 cycles. However, as described in 4.3, 3½ cycles of the 5 cycle digital processing time is due to a ripple-removal filter. When a voltage transient of greater than 0.005pu occurs, this filter automatically detects the transient and passes the measured signal straight through un-filtered. Thus, during transient events, the latency of the total measurement system drops from 5.54 to 2.04 cycles.

For the most time-critical measurements, the 1½ cycle measurement can be tapped off before the 3½ cycle ripple removal filter. This results in a measurement with a latency of about 2.04 cycles at all times, with increased ripple and noise.

This measurement architecture, which is fully implemented and tested within this thesis, allows "Class A" measurement accuracies to be achieved but at a low frame rate and with short measurement latencies. The performance is summarised in Table 4-8 below.

	BS EN 61000-4-30 "Class A" performance	Performance of architecture developed in this thesis, normal scenario	Performance of architecture developed in this thesis, worst case scenario
Scenario	Harmonics up to 20% THD (twice times BS EN 61000-4-3 table 5, class 3) (BSI, 2002)	THD up to 28% Instrumentation noise level (before the ADC) at or below 0.001pu RMS (60dB SNR). A 12-bit ADC with a bit noise of no more than 1 bit RMS.	THD up to 53%. Instrumentation noise level (before the ADC) at or below 0.005pu RMS (46dB SNR). A 12-bit ADC with a bit noise of no more than 2 bits RMS.
Voltage measurement	±0.1% (±0.001 pu) A measurement time of 10 cycles (200ms), assuming steady state frequency	accuracy/ripple of ±0.0011pu/±0.0010pu respectively. A settling time of 2 cycles (40ms) during transients and 5.5 cycles (110ms) normally. Voltage amplitude error widens to ±0.003pu with ROCOF rates of ±10Hz/s, and proportionately with lower ROCOF rates.	accuracy/ripple of ±0.0028pu/±0.0023pu respectively. A settling time of 2 cycles (40ms) during transients and 5.5 cycles (110ms) normally. Voltage amplitude error widens to ±0.003pu with ROCOF rates of ±10Hz/s, and proportionately with lower ROCOF rates.

Table 4-8 : Performance of voltage amplitude measurement architecture proposed by this thesis, versus standard "Class A" performance

All but one of the original requirements for the measurement of fundamental amplitude and phase (section 2.10, Table 2-11 and Table 2-12) have been fully met by the algorithms presented in this chapter, for even the worst cases of harmonic content and instrumentation noise, with frame rates down to 10 samples per cycle. The hardest

specification to meet is the requirement for low ripple/noise on the Fourier measurement of amplitude, for measurements with 5-cycle latencies which are used for control purposes. The largest set of constraints on meeting this specification has been shown to be high harmonic content (THD above 28%) which might appear (even transiently), combined with low sample rates inside the main processing algorithms. Noise and ADC quantisation are also subsidiary contributors to the error ripple.

The original requirement was set at ±0.001pu (Table 2-12), which is derived from the desire to keep reactive power output ripples to within a ±0.01pu ripple range, where the reactive power ripples are caused by voltage measurement ripple, with a 10% voltage droop slope. Alongside this requirement, the original hardware constraints proposed in Table 2-12 were a main sample rate of 10 Sa/s, instrumentation noise at 0.005pu RMS (46dB SNR), and 2 bits of ADC noise. In this scenario, the algorithms presented give ripple of the order of ±0.0023pu (Table 4-6) for waveforms with THD_V up to 53%. This would not be disastrous and could certainly be tolerated for short periods of time.

If the input waveforms have a THD_V up less than 28%, combined with a small improvement in the instrumentation performance, then the ±0.001pu target is met (Table 4-7 & Table 4-8) and the architecture accuracy/ripple then meets the requirements for Class A performance, but with substantially lower measurement latency than Class A allows, and using a very low frame rate.

The phase measurement errors are within 0.18° (Table 4-6) for all scenarios up to 53% THD_V, and within 0.08° (the original requirement of Table 2-13) for the improved scenario with THD_V up to 28% (Table 4-7). This measurement of phase can therefore be used as the basis for a robust frequency measurement. This is investigated fully in chapter 5.

4.8   References for chapter 4

BSI, British Standards. (2000). Voltage characteristics of electricity supplies by public distribution systems. BS EN 50160:2000. ISBN 0 580 35925 5

BSI, British Standards. (2002). Electromagnetic compatibility (EMC) — Part 2-4: Environment — Compatibility levels in industrial plants for low-frequency conducted disturbances. BS EN 61000-2-4:2002. ISBN 0 580 40489 7

BSI, British Standards. (2003). Electromagnetic compatibility (EMC) – Part 4-30: Testing and measurement techniques – Power quality measurement methods. BS EN 61000-4-30. ISBN 0 580 42081 7

University of Strathclyde

Department of Electronic and Electrical Engineering

Measurement, control and protection of microgrids at low frame rates supporting security of supply

Volume 2

Andrew Roscoe

A thesis presented in fulfillment of the requirements for the degree of Doctor of Philosophy

February 2009

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has lead to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed :   Date :

5   Frequency measurement

Within microgrids, as already described in section 2.1, system frequency is a much more dynamic parameter than it is in large national networks. This is due to the lower per-unit inertia of the generators and spinning loads, in combination with the larger sizes of commonly occurring load steps, relative to the system capacity and inertia. Thus, a measurement of frequency must respond fast enough to accurately follow the system frequency variations. The requirements for measurement accuracy, tracking and latencies were previously discussed section 2 and culminated in Table 2-14 and Table 2-15. This specifies a measurement latency of only 5 cycles (100ms) and a required accuracy of ±0.025Hz (±0.0005pu), with only 10 samples per cycle, and other interfering quantities such as harmonic content and flicker. The measurement of frequency to this accuracy with such a short latency is a specialised problem, to which "normal" techniques cannot be applied. In other fields of engineering, frequency measurement can generally take place over many cycles which allows much more conventional and accurate techniques to be used (Roscoe, 2005).

In addition to the requirement for a fast response time, tough requirements for "influence qualities" such as high harmonic content, power-line communications, flicker, and unbalance have all been laid down in section 2.7. The expected level of these interfering parameters is much higher within microgrids than it is for "normal" grid-connected power systems. This will be especially true for a battlefield and disaster-relief type scenarios when small power system networks might be operated outside the normal design constraints (and legal requirements) which would be applied for a distribution network. In such cases the risks of the following conditions are all increased: lower diversity of loads potentially injecting common harmonic currents, lower fault levels (higher impedances) causing higher levels of voltage harmonics due to harmonic currents, higher unbalance due to the difficulty of balancing single phase loads accurately across three phases, and large relative load steps relative to the size of the network leading to flicker and frequency deviations.

In addition to all these constraints, it is also desired to measure frequency using cheap equipment and processors, at relatively low sample rates so that the processor can also be used for other control/relaying tasks simultaneously (see section 2.9).

This chapter presents the design and performance of a new algorithm for frequency measurement. It can be called a "Clarke-FLL hybrid", being a combination of two entirely

separate frequency measurement algorithms based upon (firstly) a Clarke transformation and (secondly) a set of 3 single-phase frequency locked loops (FLLs). Such a hybrid is created because the two different measurement techniques have different strengths and weaknesses. The hybridisation enables the overall algorithm to exploit the strengths of both (and minimising the weaknesses) by selectively using one or the other depending upon dynamic circumstances. The algorithm provides measurements of frequency, amplitude and phase of a 3-phase signal set. The pre-existing work which directly contributes to the algorithm is the well-established mathematical Clarke transformation itself (Clarke, 1943), and some of the exact-time averaging and Fourier measurement blocks within the MATLAB SimPowerSystems blockset which were significantly enhanced in sections 3.2 and 3.5. The single biggest influence on the core of the final measurement design is the finding that the use of the optimised exact-time averaging techniques (particularly the novel extra ½-cycle averaging introduced in section 3.9) enable the Clarke transformation and Fourier algorithms to give significantly better results than other authors have previously achieved, even at very low sample rates. The work by Jovcic (2003) was of significant interest, both because he is the only other author to date to recognise the benefit of exact-time averaging, and also due to his use of 2nd harmonic cancellation techniques (although these ultimately did not contribute to the final algorithm, as described by section 3.10).

The final Clarke-FLL hybrid contains a large amount of Simulink code. This consists of many carefully placed uses of the exact-time averaging algorithm (optimised in section 3.2), the novel ripple remover (designed in section 4.3), and a significant quantity of code which implements:

- fault ride-through

- fast settling when a new or fast-changing signal is applied, including seeding of the FLL with data from the Clarke's measurement when appropriate

These two functions contain large amounts of code to implement logic, buffering, and hysteresis delays. All this code was created from scratch during this work.

In the section 5.7, the performance of the Clarke-FLL hybrid is trialled against 7 other candidate frequency measurement systems. It is shown to perform the best, and is thus the solution of choice. Benchmarking analysis is also performed to show that the computational load presented by the Clarke-FLL hybrid is competitive compared to the other solutions, despite is apparent complexity.

5.1   Established and published methods

Within a nationwide power system, adequate and accurate frequency measurement has traditionally been made with relatively simple techniques such as zero-crossing detection. These techniques have been adequate due to a combination of factors:-

- Smaller numbers of frequency measurement points required means that equipment can be relatively expensive
    - o   fast sample rates, sometimes into the 20 kSa/s range
    - o   precision analogue hardware front-ends
    - o   precision ADCs (analogue-digital converters)
- averaging over many cycles, which has been acceptable because the expected rate of change of frequency (ROCOF) has been low
- reliance upon the sinusoidal nature of the voltage waveforms
- reliance upon balanced operation, so that only one phase needs to be measured

In the following sub-sections, a summary review of established and proposed methods for frequency measurement within AC power systems is presented. Some of the proposed methods use novel techniques. However none (in their presented or raw implementations) are found to meet the requirements of chapter 2.

5.1.1   Frequency measurement by zero crossings

Traditional commercial relays have tended to use zero crossing algorithms to measure frequency. By taking samples, it is possible to find the pairs of samples which contain the zero crossing. Linear interpolation between these two samples can then be used to calculate the time at which the zero crossing is believed to have occurred. It is interesting to perform an error analysis for this frequency measurement method, using a purely sinusoidal waveform input, as a benchmark against which to analyse other frequency measurement methods and scenarios.
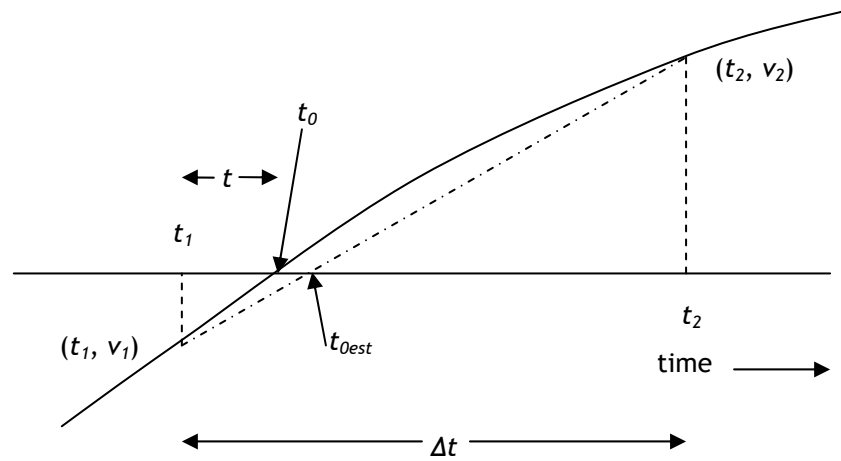
Figure 5-1 : Linear interpolation to estimate the time of a zero crossing

At a zero crossing point, the actual zero crossing occurs at time t_0. The sample interval is Δt_s. The first sample v_1 occurs at $t_1=t_0-t$, and the second sample v_2 occurs at $t_2=t_0-t+\Delta t_s$. If one of the sample values v_1 or v_2 is 0, then by examination the actual position of the zero crossing time is known to be at t_1 or t_2 respectively. Normally, however, this is not the case, and linear interpolation is carried out to determine an estimate of t_0, given by t_{0est}.

The estimate of t_0 can be made using the following equation:

$$t_{0est} = t_1 - v_1 \left(\frac{t_2 - t_1}{v_2 - v_1} \right)$$ where v_1 and v_2 are the signed voltages (v_1 negative and v_2 positive

in Figure 5-1)

(5.1)

The effect that this single zero-crossing measurement error will have on a frequency measurement at 50Hz is given by:-

$$F_{error} = \frac{1}{T} - \left(\frac{1}{T + (t_{0est} - t_0)} \right)$$ where T is the actual period of the sinusoid.

This expression simplifies, assuming that t_{0est}-t_0<<T, to:

$$F_{error} = \frac{(t_{0est} - t_0)}{T^2} = (t_{0est} - t_0)F^2$$ where F is the actual frequency

(5.2)

It is relatively trivial to now analyse the frequency error induced by the error t_{0est}-t_0 for different values of sample interval Δt_s and different values of the time offset t between the first sample and the zero crossing. At 50Hz, and using a 1 per-unit voltage magnitude,

v_1=sin(2π50(t_1-t_0)) and v_2=sin(2π50(t_2-t_0)). These values for a pure sinusoidal waveform can be substituted into (5.1) for different values of Δt_s and t to evaluate t_{0est}-t_0. For simplification, t_0 may arbitrarily be set to 0 for this evaluation.
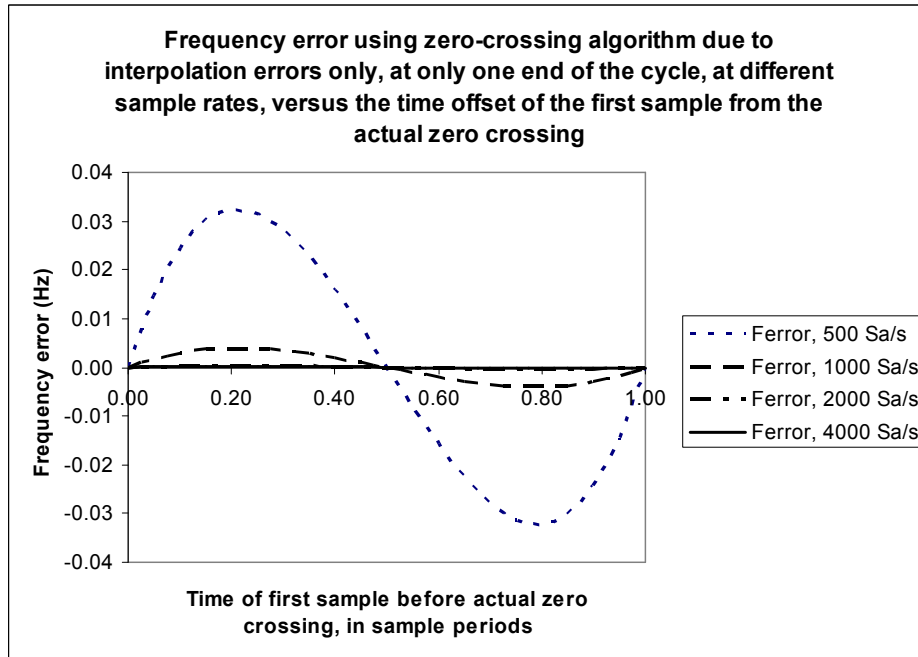
Figure 5-2 : Frequency error of zero crossing algorithm using pure 50Hz sinusoid inputs

Clearly, from Figure 5-2, the zero crossing algorithm doesn't work well at sample rates below 1000 Sa/s (20 Sa/cycle), when the sample instants do not coincide with the actual instants of the zero crossings. This is the reason that most traditional digital frequency measurement devices use one of the two following approaches:-

a) a very high sample rate, above 4 kHz and reportedly as high as 24 kHz (480 Sa/cycle), and/or

b) front-end sampling hardware which is clocked at a variable rate by a phase-locking system which targets samples at the exact instants of zero crossings. In this case, the performance of the hardware phase-locking PLL system will also affect the accuracy and dynamic response of the measurement.

Aghazadeh (2005) presents a method of reducing the errors, compared to the analysis performed above, for a zero-crossing measurement. However, the resulting frequency errors are ±0.05Hz when representative noise and harmonics (commensurate with the levels applied in this thesis) are included on the input signal. This is an unacceptable error level.

5.1.2   Methods based upon the Clarke transformation

An appealing method for measurement of system frequency in a 3-phase system is the use of the Clarke transformation (Clarke, 1943) to transform the three-phase voltage measurements into a rotating vector in 2-dimensions, by the standard equation:-

$$
\begin{bmatrix} A \\ B \\ 0 \end{bmatrix} = \begin{bmatrix} \dfrac{2}{3} & \dfrac{-1}{3} & \dfrac{-1}{3} \\ 0 & \dfrac{1}{\sqrt{3}} & \dfrac{-1}{\sqrt{3}} \\ \dfrac{1}{3} & \dfrac{1}{3} & \dfrac{1}{3} \end{bmatrix} * \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}
$$

(5.3)

The angle of the vector $\begin{bmatrix} A \\ B \end{bmatrix}$ rotates at the system frequency. Thus, system frequency at any instant can be defined mathematically as

$$
f = \frac{\partial}{\partial t}\big(\arctan 2(B, A)\big)
$$

(5.4)

Where arctan2(B,A) is the well-known version of arctan that returns a correct answer over the full 4 quadrants of the plane, and also avoids division by zero errors. Also, a subtlety is that when arctan2(B,A) crosses the boundaries at ±π, 0 or 2π, an algorithm must correctly "unwrap" the phase to give a correct answer for f. Several authors make use of this technique to measure frequency.

Cantelli (2006) uses a version of the Clarke transform method, specifically because it can ride "seamlessly" through dips on single phases, whereas a single-phase frequency measurement is vulnerable to dips on that phase. The problems of phase unwrapping are avoided because the method does not use the atan() function but instead estimates the phase change from the Cartesian coordinates. There is a small time saving by avoiding the atan() function, although the evaluation does require a sqrt() function instead (see Fig. G-1 for relevant benchmarking experiments). The method is also vulnerable when sample rate is low, as the approximation to the atan() function then becomes less valid. Cantelli's method does not carry out exact-time averaging over ½ or one cycle, and so there are ripples at the output of his algorithm in the presence of harmonics or unbalance. This is reduced by a PID filter which of course introduces an IIR response characteristic. Cantelli (2006) ignores the effects of unbalance, which is a significant oversight, particularly as

unbalance can reach 100% during an unbalanced fault. No mention is made of the sample time used in any of the analysis. From the graphs presented in the paper it appears to be >>10 samples per cycle.

5.1.3   Methods based upon Fourier transforms

Aside from zero crossings, Fourier transforms are the most obviously suitable ways to measure the frequency of a sinusoidally shaped waveform, from a purely mathematically point of view. This is because they correlate the measured waveform with a synthesised sinusoidal waveform, in the real or complex domain. The transforms can take the form of an individual discrete Fourier transform (DFT), carried out for one exact frequency correlation (as used in this thesis), or as a repeated set of transforms at numerous correlation frequencies to provide an array of data across a frequency range. The FFT (Fast Fourier Transform) is a specialised form of this, which reduces the computational effort but imposes restrictions on the input and output data streams (regular sample times and fixed output frequencies). A single DFT calculation requires (many) sine and cosine evaluations, while an FFT calculation required many floating-point operations. Due to the computational effort required to calculate the DFT/FFT, such methods have taken some time to become acceptable within power systems applications. Instead, such methods as zero crossings and Walsh functions (Johns, 1995) have been used.

More recently, microcontroller CPU speeds (and particularly the floating-point capabilities) have gradually advanced to an extent that Fourier analysis can be carried out directly, so long as care is taken within the coding. Early digital implementations of Fourier analysis within power systems relays were simplified within the algorithms to use such things as sine/cosine lookup tables, or lookup tables for an entire sine/cosine correlation waveform at a fixed sample rate (Moore, 1996a). This type of implementation suffered from ±0.04Hz errors for input signals off-nominal frequency, due to errors resulting from limitations of lookup tables. An example of a very poorly coded Fourier analysis can be seen in Lin (2005). At the present time, analysis of Appendix F & Appendix G shows that sine and cosine operations can be calculated within modern microcontrollers almost as quickly as a lookup table can be accessed. This, particularly when combined with the realisation of exact-time averaging algorithms (see section 3.2) which can provide rolling-buffer operation, means that much more effective and accurate Fourier analysis algorithms can be coded in real-time using cheap microcontrollers.

5.1.4   Phase-locked loops (PLLs)

The MATLAB Simulink SimPowerSystems blockset provides code which implements both single and three-phase PLLs. These perform reasonably well, as they contain single-cycle

exact-time averaging. They are used for 2 of the 8 methods compared in trials presented later in this thesis.

Chung (2000) presents a quite conventional digital three-phase PLL using the dq frame, without any special techniques such as signal level normalisation or exact-time averaging. The resulting output suffers ripple in the presence of unbalance and harmonics. No mention is made of the sample time used in any of the analysis. From graphs it appears to be >>10 samples per cycle.

Awad (2005) describes a software PLL system. He introduces a useful normalisation of the [AB] vector to unit length before carrying out the phase detection, which in a PLL can be done by looking purely at the magnitude of the q component after transformation to the dq frame. This normalisation helps the PLL to retain the same dynamic performance during voltage dips. This normalisation, however, would not be required if the atan2(q,d) function was used to measure the actual phase rather than the estimated phase. Awad (2005) also uses a delay-signal cancellation technique to separate out the positive and negative sequences of the AB vector, so that the transformation to the dq frame contains only the positive sequence component. This is done, presumably, to minimise PLL ripple due to unbalance (but not harmonic content). No mention is made of the sample time used in any of the analysis. From graphs presented it appears to be >>10 samples per cycle. The delayed-signal cancellation is an approximation which will not be correct at all times, and assumes that system frequency is always at or very close to nominal. In a microgrid scenario, this is not necessarily true. The delayed-cancellation technique in Awad (2005) would not be required if an exact-time averaging algorithm over exactly ½ or 1 cycle was applied to the dq data. This would remove all ripples due to unbalance, and also harmonics.

Han (2006), presents a single-phase PLL, which adapts the sensitivity of the phase detector by using an ALC (normalisation) control loop so that during voltage dips the PLL retains the same gain. This concept is also used (but in a more comprehensive manner) in a PLL designed by Jovcic (2003).

Jovcic (2003), presents a PLL which does address the issue of low sample rate. This is done by the use of the SimPowerSystems block "Discrete Variable Frequency Mean value". Jovcic (2003) also introduces the concept of 2nd harmonic cancellation. These concepts have been analysed and extended in sections 3.2 and 3.8 of this thesis. Jovcic's PLL is a single-phase device, although the concepts extend readily to three-phase operation. This

PLL has been recreated in detail, exactly as described by the paper. During this work it was found by simulation that below 2000Sa/s the feedback loop inside the algorithm for voltage magnitude estimation can become unstable under high levels of harmonic contamination. At sample rates of 2000Sa/s and above, the PLL operates as published.

Jovcic's PLL therefore includes several useful techniques, and the performance of the phase-locked loop itself is very good. However, being a PLL, the author proposes that it is not the best way to measure frequency (see section 5.3). Also, the algorithm does not include any fault ride-through mechanisms (see requirements section 2.7.5). This thesis proposes (but does not explicitly prove) that if the frequency measurement algorithm described later in section 5.4.2 were used in combination with a PLL such as Jovcic's, the resulting algorithm would outperform the Jovcic PLL as published. This is because the frequency measurement algorithms proposed here can settle more quickly without overshoot, ride through faults better, and attenuate noise better, than the phase-tracking part of a PLL. The output of a better frequency measurement (not from a PLL) could in future be used to drive the exact-time averaging algorithms inside a PLL phase detector, to create a hybrid frequency measurement and PLL algorithm.

5.1.5   Kalman filters

Dash (2000) proposes using extended Kalman filters. This thesis does not propose to utilise Kalman filters due to their similarity to low-pass filters and the inferior rejection of noise and ripple offered by these compared to adaptive FIR filters (see section 3.3). The algorithms proposed by Dash have not been finalised and appear to be unstable under certain conditions. The graphs of results are also scaled in such a manner that the errors cannot be resolved to useful accuracy. No mention is made of the sample time used in any of the analysis. From graphs presented it appears to be >>10 samples per cycle.

5.1.6   Other methods

Choi (2006), attempts to use a phase-delaying technique on a single-phase voltage waveform. This is done to create two voltages from one, with, hopefully, one lagging 90 degrees behind the other. This can be processed in a similar way to the AB vector in Cantelli (2006). No mention is made of the sample time used in any of the analysis, and the diagrams suggest the system has actually been modelled in the "continuous" mode.

The results show ripple of the order of ±1Hz which is unacceptable. The delayed signal technique is far worse at removing unwanted effects of harmonics than exact-time averaging.

Salcic (2000) presents an extremely convoluted method which does not assume that the waveform is sinusoidal but instead simply looks for symmetry around the positive peaks, and then attempts to fix a timestamp at each peak. The time between timestamps is then used to deduce frequency, like a zero crossing algorithm. The sample rate requirements for this method are $Fmax.(Fmax/Fresolution)$ = 53*53/0.1 = 28090Hz to achieve even a 0.1Hz resolution, which is unacceptable within the scope of this thesis. Waveforms with substantial THD (35%) are used as test waveforms. However, the resulting errors even with 28kSa/s sample rate, over 10 cycles, are still about 0.01Hz. This is poor performance for such a high sample rate. With a 1 cycle measurement, the errors are about 0.08 Hz. The algorithm also appears to have a quantised output; i.e. the output for the 1-cycle case is 49.925Hz, 50Hz or 50.075Hz, and cannot take values in between. Two fundamental problems within this paper are that the algorithm is looking for a maximum in a waveform, which is always numerically inaccurate compared to locating a zero, and that the algorithm appears unable to interpolate between samples so the sample rate required becomes unnecessarily large.

Lin (2005) describes an interesting algorithm (for both amplitude and frequency measurement) based upon adaptions of wavelet transforms. The sample rate used is also suitable, at 600Sa/s, and this method might be worth more analysis in the future. No noise or ADC quantisation is applied, however, and the scales on the graphs do not allow a good analysis of the performance. His proposed method is compared to a Fourier analysis (which is the proposed base measurement used in this thesis), but this appears to have been deliberately coded very poorly to give bad results for comparison.

Lopez (2008) describes a 9-sample technique (triplet of triple samples), which claims to be fast responding. However, similarly to the 2 and 3-sample amplitude measurement techniques (Johns, 1995), the method is very susceptible to noise and ripple due to harmonics. All the dynamic performance advantage of the technique is lost because the samples must be pre-processed by a bandpass filter which introduces a delay of at least 2 cycles to achieve even marginal performance (±0.02Hz accuracy with 40dB SNR at 128 samples per cycle). The technique has been coded in Simulink by the Author and improved with averaging techniques to create a 5-cycle measurement and compared to the proposed method of this thesis (subsequently to the text of this thesis being substantially complete). The Lopez (2008) method compares poorly. In particular, in the presence of harmonics the algorithm produces biased results with an over-estimate of frequency which cannot be removed by any amount of post-filtering. Thus, the algorithm pre-filters must be able to provide almost perfect sine waves – an almost impossible task without adding substantial group delay. In test, the method over-estimated frequency by up to 0.3 Hz for high

harmonic content (28% THD) signals, even with 2 cascaded 125 Hz low-pass pre-filters.

5.2   Summary from literature search for frequency measurement

- Most published papers discuss PLLs (Phase-Locked Loops), of which the primary function is to control power-electronic devices. The estimate of frequency is a by-product of this process.

- Few authors appear to fully prioritise the reduction of sample rates to enable the algorithms to be combined in multi-function systems. Exceptions are Lin (2005) who uses a sample rate down to 600 Sa/s, and Jovcic (2003), who addresses some of the problems due to lower sample rate, although the actual sample rates targeted or used is not described.

- Some authors apply harsh levels of harmonics, and some present unbalanced scenarios. However, in most cases the graphical results presented either show relatively poor errors with noise, ripple, or IIR settling characteristics, or show errors which have been hidden by graphs with wide scales.

- Most of the published algorithms can be either made redundant or significantly improved by use of an exact-time averaging algorithms. These are used by Jovcic (2003), and have been significantly extended/improved in this thesis (see sections 3.2 and Appendix G). Use of these blocks allows removal of much of the ripple due to unbalance and harmonics, and also removes noise more effectively than a low-pass filter (see chapter 3).

5.3   Why not to use a PLL for a measurement of frequency

During the course of this thesis, many different candidate methods for frequency measurement were investigated. Most were eventually rejected, and not all are presented here. Significant time was taken creating phased-locked-loop (PLL) solutions, not least because a 10kVA 3-phase grid-tie inverter with islanding capability was designed, built, coded and tested by the author. These phased-locked-loops were either single-phase units or three-phase dq analyses. Generally the algorithms, at the core, were very similar to the SimPowerSystems blockset algorithms "Discrete 1-phase PLL" and "Discrete 3-phase PLL". Variations on these algorithms which have been developed include:-

- Normalisation of the dq vector to unity magnitude, to standardise and stabilise the response during voltage dips/surges.

- Replacement of the phase detector (the measured value q from the dq vector) in the three-phase unit, with an atan2(q,d) result which allows faster locking or

re-locking when phase error is significant. This helps particularly when the dq vector crosses into the left quadrants of the dq plane, i.e. when d is negative and the PLL is significantly unlocked.

- Enhancement/improvement of the exact-time averaging algorithms as detailed in section 3.2.

- Experimentation with other post-PLL smoothing techniques for the estimation of frequency. (SimPowerSystems blocks use a slew-rate limit of ±12Hz/s combined with a 2nd-order low-pass filter set to 25Hz cut-off).

- Creation of PLLs with ½-cycle base correlations, and N-cycle base correlations. These allow faster response with more noise, or slower response with smoother outputs.

- Addition of ride-through capability during faults (particularly relevant for inverter drive PLLs)

- Additions which allow the PLLs to be seeded from an external estimate of frequency. The external estimate should be fast responding and robust, but can be noisy. This allows the PLL range to be extended over more than an octave which is unconventional. A seeding algorithm detects when the PLL frequency is substantially different to the robust seed frequency (with additional time qualifications), and when necessary resets the PLL to the seed frequency to re-lock it quickly. This allows extended frequency range since accidental locking to sub-harmonics or harmonics is no longer possible. It also significantly speeds up initial locking when a signal is first applied and the PLL has previously been free-running with a noise-only input.

Despite all the work on PLLs, in particular the seeding algorithms, this thesis proposes that they are not the best way to measure the parameter frequency. They lock to phase very well, which is what they are fundamentally designed to do. Because they lock to phase, however, transients cause an IIR response through the PI controller, and a step change in frequency always results in a response which overshoots. To demonstrate this, a simple Simulink simulation was created. This contains several PLLs:-

- SimPowerSystems 1-phase PLL

- The PLL designed by Jovcic (2003)

- the author's single-phase PLL based upon a single-cycle averaging of the q error, including the improved exact-time averaging algorithm. This has the same loop gains as the SimPowerSystems PLL.

- the author's single-phase PLL based upon a ½-cycle averaging system, with higher

loop gains and faster response.

- the author's single-phase PLL based upon a 1½-cycle averaging system, with lower loop gains and a smoother response.

- the author's three-phase PLL, based upon a 1-cycle averaging system of atan2(d,q). This has the equivalent loop gain of the SimPowerSystems 3-phase PLL block.

These PLLs were exposed to a waveform of a constant frequency 49.95Hz, at magnitude 1pu until t=1 second, when the frequency undergoes a step change to 50.05Hz. Figure 5-3 shows the response of the different PLLs.
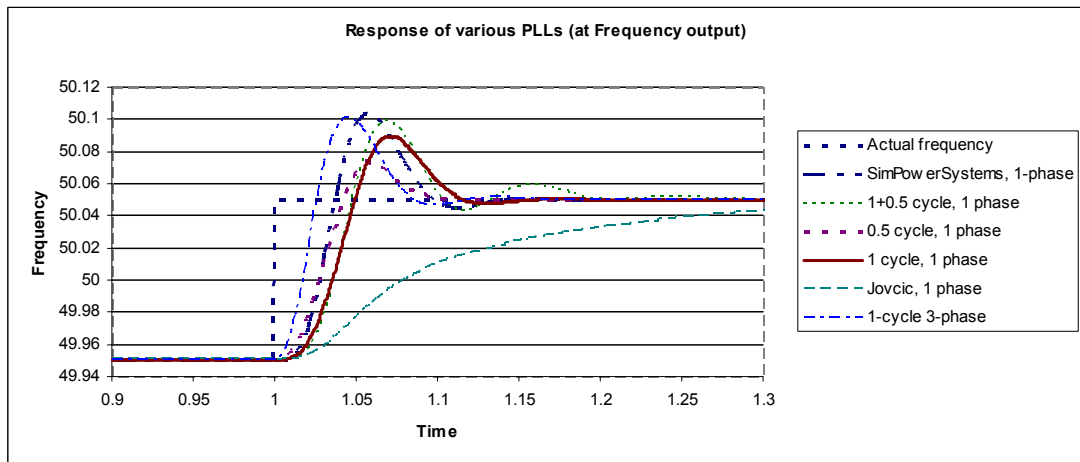
Figure 5-3 : Step response of various PLLs to a small frequency change

The responses of the PLLs exhibit two characteristics in Figure 5-3 which are undesirable. Firstly, the responses all overshoot, except for that of the Jovcic (2003) PLL, because it is so heavily filtered that its response is too slow. Secondly, the IIR response of the PI control loops means that the frequency measurement is still settling up to 200ms after the step change was applied. This is too slow to meet the requirements of 3-5 cycles (60-100ms) defined in section 2.1. It must be noted that a frequency seeding algorithm was included in the authors' PLLs for the above experiment, but since the frequency step is very small, the seeding algorithm quite rightly does not take action. This is because the frequency seeding can only seed the frequency and not the phase, so is only of benefit when the PLL has lost lock completely. This does not occur during the above simulation.

To contrast with Figure 5-3, Figure 5-4 shows the response to the same waveform of a simple measurement using the Clarke transform. In this case, the algorithm uses initial base averaging over exactly 1 cycle, cascaded into a second 2-cycle exact-time average

filter. This creates an excellent filter combination to reject noise, harmonics, unbalance and integration/interpolation errors due to low sample rates, as described in chapter 3. Clearly, the frequency measurement rises over 3 cycles (60ms) in a controlled fashion, with a closely defined FIR response and zero overshoot. This is preferable by far to any of the responses shown in Figure 5-3.
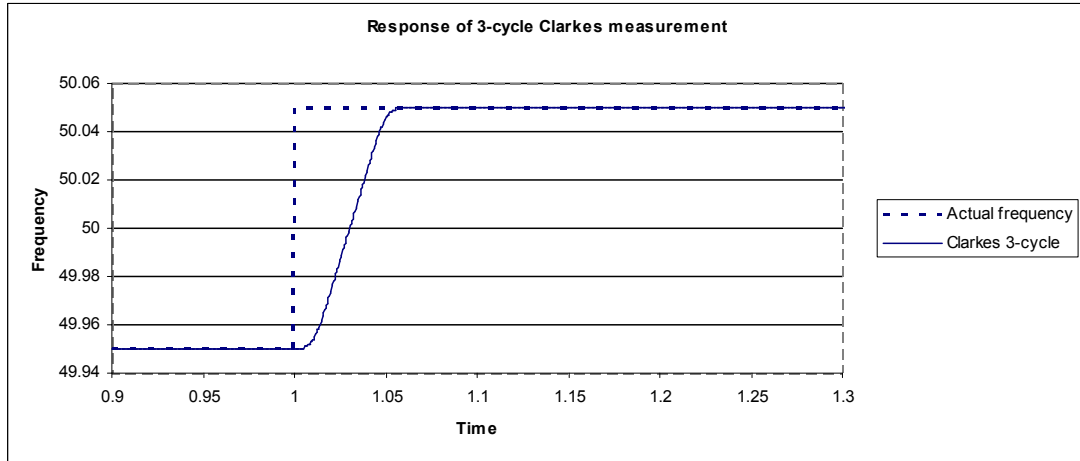
Figure 5-4 : Step response of 3-cycle Clarke transform measurement to a small frequency change

Aside from the IIR response and the overshoot characteristic, there are two more undesirable features of PLLs, when used to measure frequency:-

- Due to the control loop action, the ripples at the PLL "frequency" output may be at high frequency, but not at multiples of the fundamental frequency, even for situations where there are no aliased harmonics. This makes removal of the residual ripples much harder than for the FIR-based measurements. As shown in chapter 3, the FIR-based measurements using 1 and ½-cycle averaging create ripples at integer multiples of the fundamental for all the following cases, and these ripples can then be almost entirely removed by cascading a further exact-time average step. This almost completely removes the errors due to:-
    - o Integration/interpolation
    - o Un-aliased harmonics
    - o Unbalance (for 3-phase AB and dq-based analyses)
    - o Errors in the estimate of frequency used to dynamically set the parameters within the FIR exact-time averaging filters.
- During a phase jump, the response of a PLL is a necessary swing (undershoot followed by overshoot or vice-versa), to bring the PLL back into lock. The PLL frequency does not truly represent the waveform frequency during this time. This is the reason that PLLs such as the SimPowerSystems block and the Jovcic (2003)

218

implementation use slew-rate limiters and low-pass filters to massage the inner-loop frequency before using it as the "best estimate" of frequency. For a FIR filter which only measures frequency and is not locked to phase, the frequency measured will still contain a "blip" following a phase jump, but this "blip" is one-sided and does not contain the large overshoot-undershoot characteristic of a PLL. This is a useful feature, considering that switching large loads within microgrids can have exactly this effect due to the changing phase across a reactive component such as a transformer when real power flow changes significantly. For example, Table 2-15 explains how an almost instantaneous 10 degree phase jump could easily occur within a microgrid.

For all these reasons, the author does not believe that the use of a PLL is the best way of measuring frequency. This thesis instead proposes that techniques based upon FIR filters, without any form of PI control loop, are more appropriate.

5.4 Description of the eight methods trialled

Eight methods of estimating frequency are presented here. Six of these are described briefly, as they are used mainly for comparison. The best two methods are described first, in detail, since a combination of these two methods produces the best frequency measurement achieved to date, against the requirements of chapter 2. The test waveform used is that described in sections 2.10 & 2.11.

5.4.1 Method 1 : Clarke transformation with ride-through

The basic technique for frequency measurement using the Clarke transform was introduced in section 5.1.2. The algorithm shown below uses this technique with a number of enhancements. These allow the algorithm to switch between different modes and give fast initial settling, followed by more averaging during steady-state operation, combined with fault ride-through capability. Important considerations are pre-loading of filters during ride-through and settling, and also internal checks for validity of the final answer. These lower level details of the algorithms are all required to make this measurement as responsive as possible, and to optimise its potential for use as a seed algorithm for other frequency measurement methods. This algorithm is extremely robust and can cope with all frequency and amplitude scenarios, unless the signal amplitudes are below the noise, or there is a two-phase fault (line-line or line-line-ground). Frequency can be measured (hardware instrumentation permitting) from DC to almost the Nyquist frequency of the frame rate, without any danger of locking to sub-harmonics or harmonics. Only one "hard" maths function is required: an "atan2" function to determine the rate of change of angle of the AB vector.

Figure 5-5 (which is split over two pages) shows the highest level detail of this algorithm. Figure 5-6 through Figure 5-8 show lower-level details. Important features of the algorithm are now described:

Inside the measurement core (Figure 5-6), the Clarke transform is performed, and the rate-of-change of the AB vector angle gives the instantaneous frequency on a frame-by-frame basis. This contains noise due to all the influence qualities; in particular the effect of unbalance is to create a ripple in this measurement. Averaging over a single full cycle removes most of the ripple due to harmonics and unbalance. The effects due to Gaussian noise are still substantial, however, as the average over one cycle is the definite integral of the sample-by-sample derivative of atan2(B,A). The effect of this is that at any time, the output of the 1-cycle measurement is actually deduced from only 4 sample values, despite that fact that there may be significantly more samples than this taken during a single cycle.

The frequency can be further averaged over another 4 cycles to create a measurement with a 5-cycle response time. This can be done using a single FIR filter, or a combination of FIR filters. An analysis of candidate filter combinations was carried out, using z-domain filter analysis and bode plotting. The options considered were:-

- Single FIR average over 4 cycles

- Two cascaded averages over 2 cycles each

- ½-cycle averaging followed by 3½ cycle averaging

As previously found in sections 3.3.2 & 3.3.4, the cascaded filters provide better rejection of higher-frequency noise than the single FIR filter. In this case, the best choice is a cascaded pair of 2-cycle averages, which are seen implemented in Figure 5-6. The bode plots for the three considered filter options are shown in Figure 5-9 to Figure 5-11.

An important explanatory note regarding Figure 5-6 is the action of the "RideThrough" and "Standby" inputs. The "Standby" input is active when it appears that there is no valid set of inputs (measurements of voltages on phases A, B & C). In this case, the averaging filters are filled with the nominal frequency. This helps to minimise settling errors if a signal at nominal frequency is subsequently connected. Also, during ride-through, the filters are filled with the frequency which is assumed to be the correct frequency during the ride-through timeframe. This again helps the measurement settle if input signals re-appear at a similar frequency when a fault is cleared.

Measure the frequency of a 3-phase signal, via Clarkes with ride-through

This is a very robust algorithm:-
- capable of measuring down to DC (hardware input filters allowing)
- capable of measuring up to 1/(3*Tps)
- Never gets stuck on harmonics or subharmonics
- Rides through short faults
- Can ride through sustained single-phase faults but NOT two or three-phase faults (line-line or line-gnd)

Always provides about the right answer unless there is a 2-phase or 3-phase hard fault.
The fully processed output is averaged for smoothing, and also rides through voltage disturbances for the allowed time.
Other outputs have less averaging and can be used for noisier, quicker measurement.

Andrew Roscoe, 2007

```
Hard math function count:-
sin/cos/tan  : 0
asin/acos    : 0
atan,atan2   : 1
sqrt         : 0
x^y          : 0
e^x, 10^x    : 0
ln, log10    : 0
```
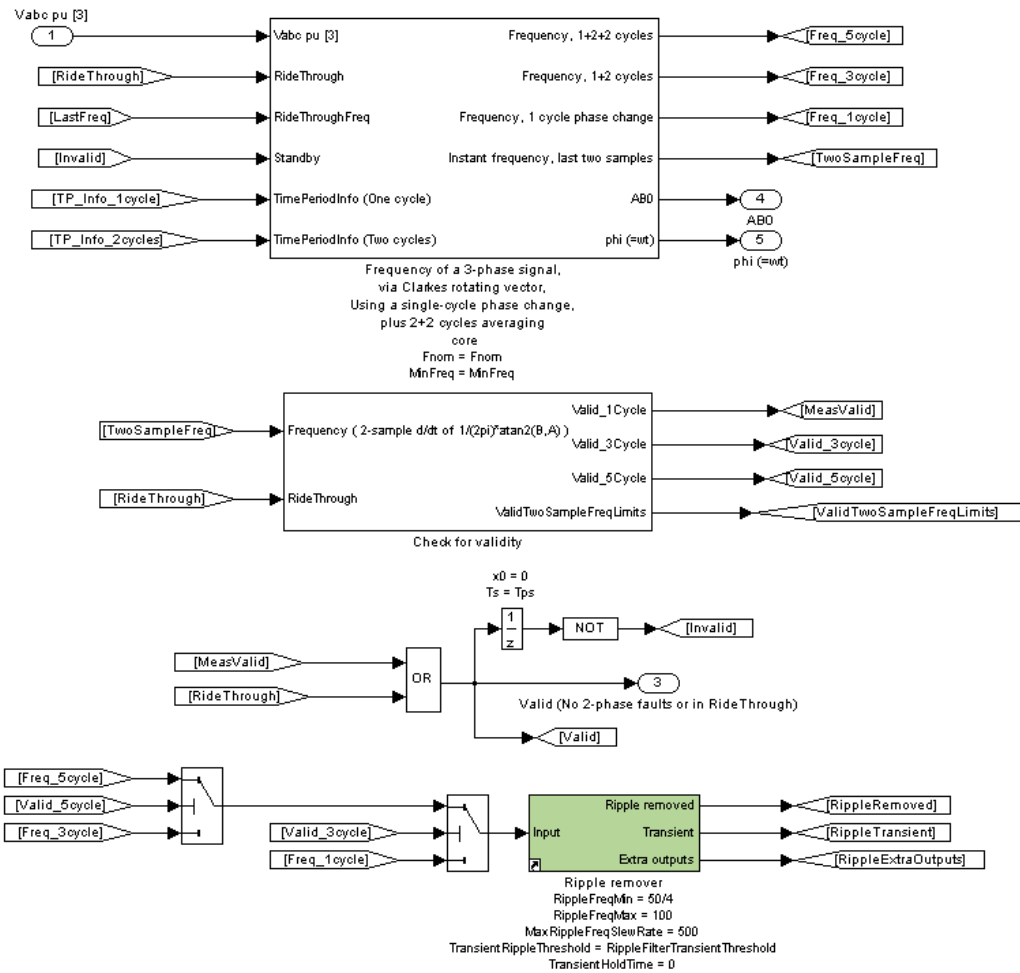

Figure 5-5 : Clarke transform frequency measurement algorithm - detail (1) [this and next page]
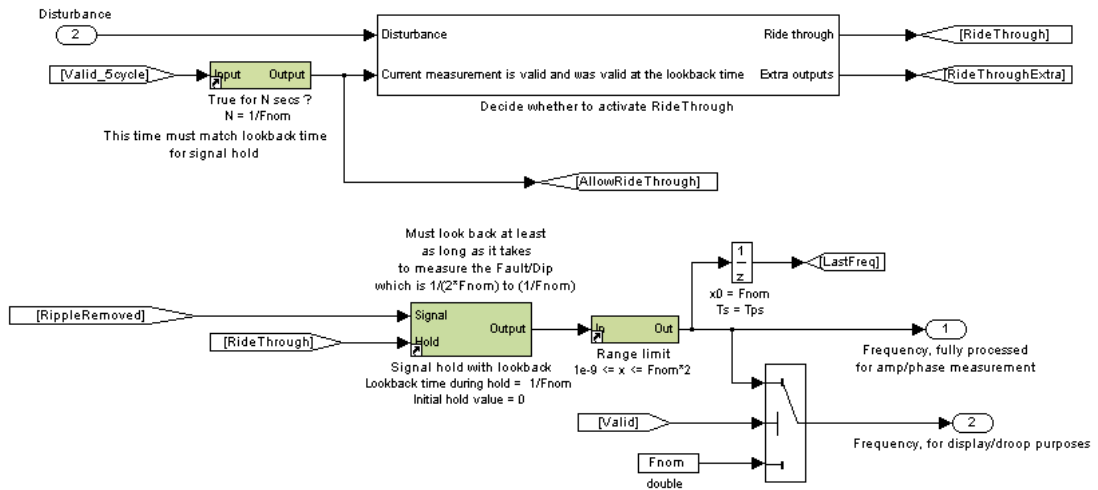
221
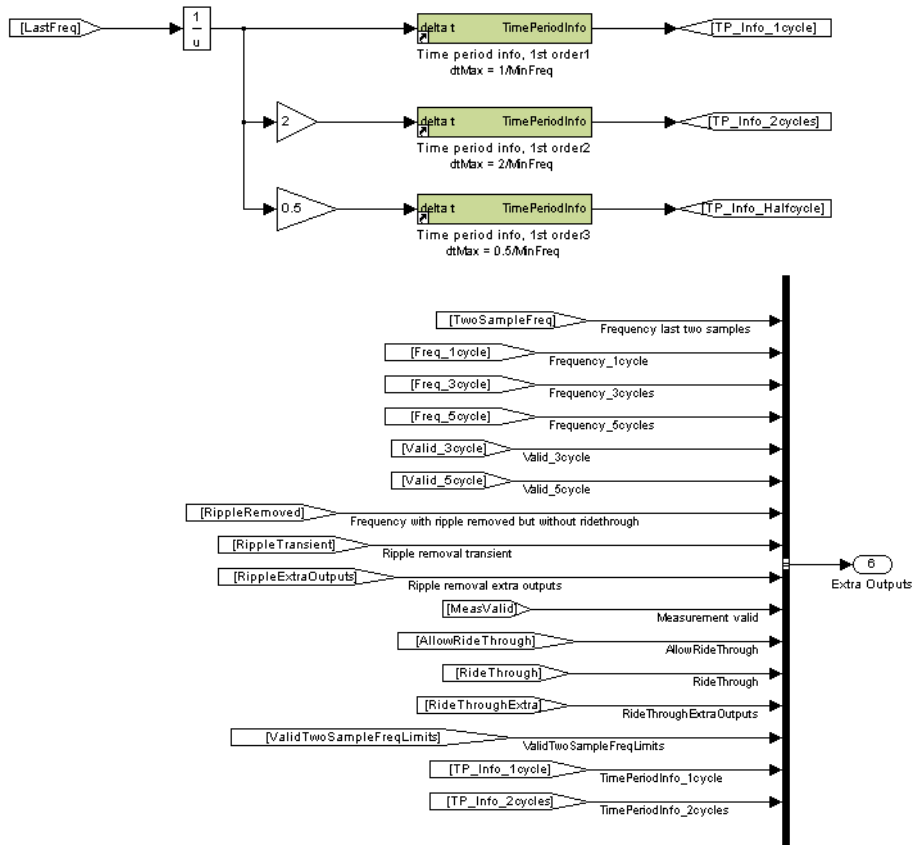
Figure 5-5 : Clarke transform frequency measurement algorithm – detail (1) [this and previous page]

Measure the frequency of a 3-phase signal,
using Clarkes transformation to AB0 followed by
measurement of the AB vector rotation over one sample,
then additional averaging to reduce
ripple at n*fundamental frequency and noise

Andrew Roscoe, 2007

Hard math function count:-
sin/cos/tan : 0
asin/acos : 0
atan,atan2 : 1
sqrt : 0
x^y : 0
e^x, 10^x : 0
ln, log10 : 0
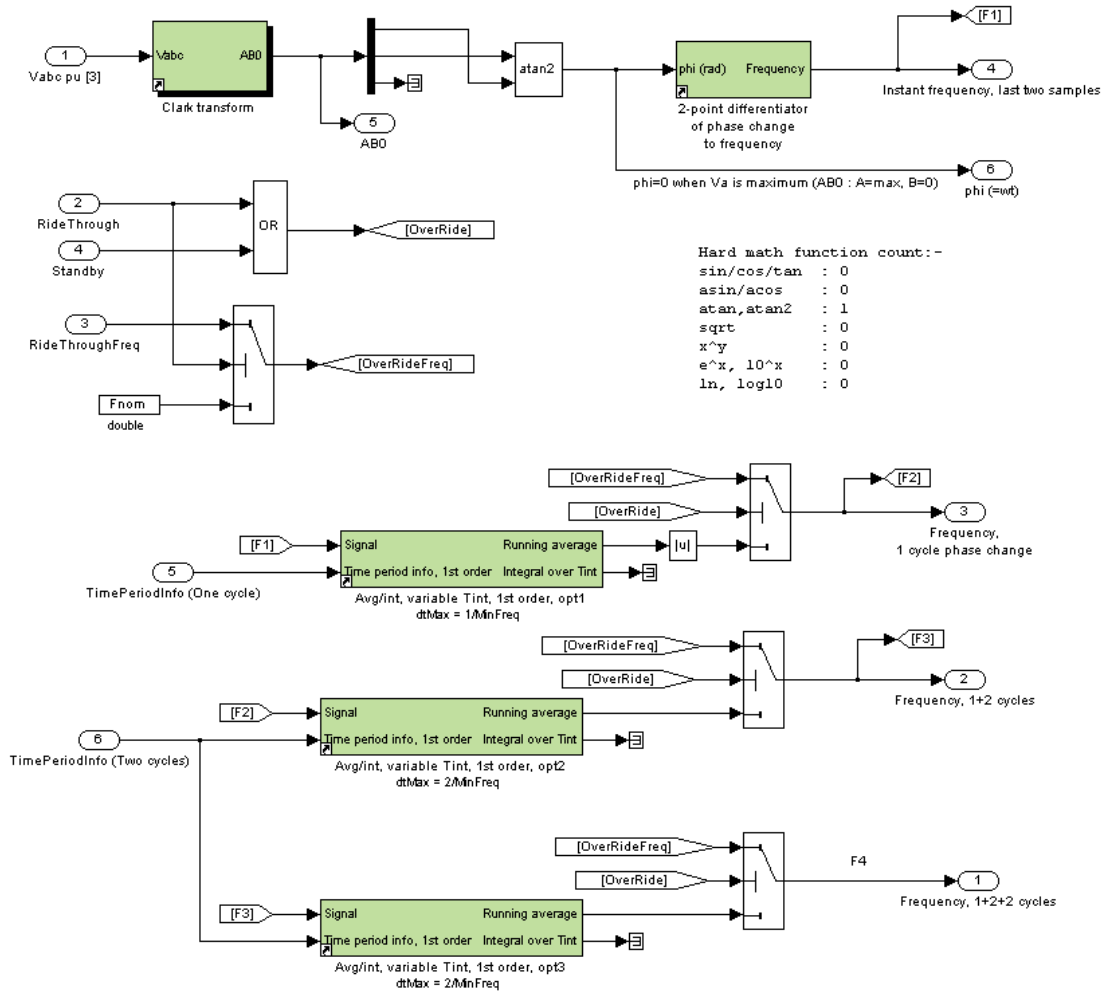
Figure 5-6 : Clarke transform frequency measurement algorithm – detail (2); core

The next stage (Figure 5-7) is to determine the validity of the measurement. It was considered to use a trajectory analysis of the [AB] vector, as derived by Ignatova (2005), for this. However, this analysis as presented takes at least a cycle to compute, which is too long. Instead, the analysis of validity is done by analysis of the instantaneous frequencies on a sample-by-sample basis, and checking these values against lower and upper limit lines which will not be exceeded when measurable 3-phase signals are applied which do not have 2-phase faults. The instantaneous frequencies are measured by the rate of change of the angle of the AB vector over a single sample frame. Either one of the following two problems will cause the limits to be exceeded:-

223

- The signal magnitudes are at a level approaching or lower than measurement noise.

- A two phase fault. This causes a collapse of the AB vector trajectory from a circle to a one-dimensional line which has a un-measurable rate of change of angle. Note that a single phase-ground fault results in a collapse of the AB vector trajectory from a circle to an ellipse of aspect ratio 1/3rd, with a maximum rate-of-change of angle per second of 3*2π*f, where f is the frequency. This is shown in appendix B.1.4.

Therefore, sensible bounds on the two-sample frequency can be set at approximately:-

Lower limit: -atan(y/0.05)/(2*π*T_s)

Upper limit: atan(y /0.05)/(2*π*T_s) + f_{nom}*1.2*3

These limits allow for

- Declaration of invalidity when the [AB] vector magnitude falls below 0.05pu, with worst case measurement noise of y pu (set to typically 4 times the RMS measurement noise which includes instrumentation noise and ADC quantisation/noise errors)

- single-phase faults at up to f_{nom}*1.2, i.e. 60Hz for a 50Hz system.

Figure 5-7 shows this checking stage. Importantly, the resulting flag "ValidInstant" is then qualified by periods of 1.5 (nominal) cycle periods, 5 cycles, and 8 cycles. These qualified flags then describe the validity of the 1, 3 and 5-cycle averaged measurements. The additional qualification time allows for settling in the hardware low-pass filters, the internal averaging stages, and also transient settling within the power system itself. These flags are subsequently used (Figure 5-5) to select the appropriately averaged frequency result to use; the fastest during "startup", or the most averaged during steady-state operation.

An alternate novel method was investigated to detect validity and the presence of two-phase faults, in a manner similar to that of Ignatova (2005). This measured the area of the ellipse enclosed by the AB trajectory. This method proved to be more complex and less effective than the above method, and is not presented here.

The chosen result is then passed through the ripple removal filter, which was introduced in section 4.3.1. This can apply a maximum additional averaging period of 4 nominal cycle periods, but during fast frequency changes it will bypass itself to provide a faster result. Note that during large transient events, the ripple removal filter must remove itself to

meet the overall response specification of 5 cycles. This is because the previous averaging stages take 1+2+2=5 cycles already, and so an entirely new frequency input will take 5 cycles to filter through the 5 cycles of prior averaging and settle. However, during steady-state and slow frequency ramps, a total averaging of 1+2+2+4=9 cycles produces a result which represents the frequency at the midpoint of this averaging time period; i.e. 9/2=4.5 cycles prior to the output result. Thus, during steady state or slow frequency ramps, the ripple removal filter can switch in and the overall response time will be just within the specifications.
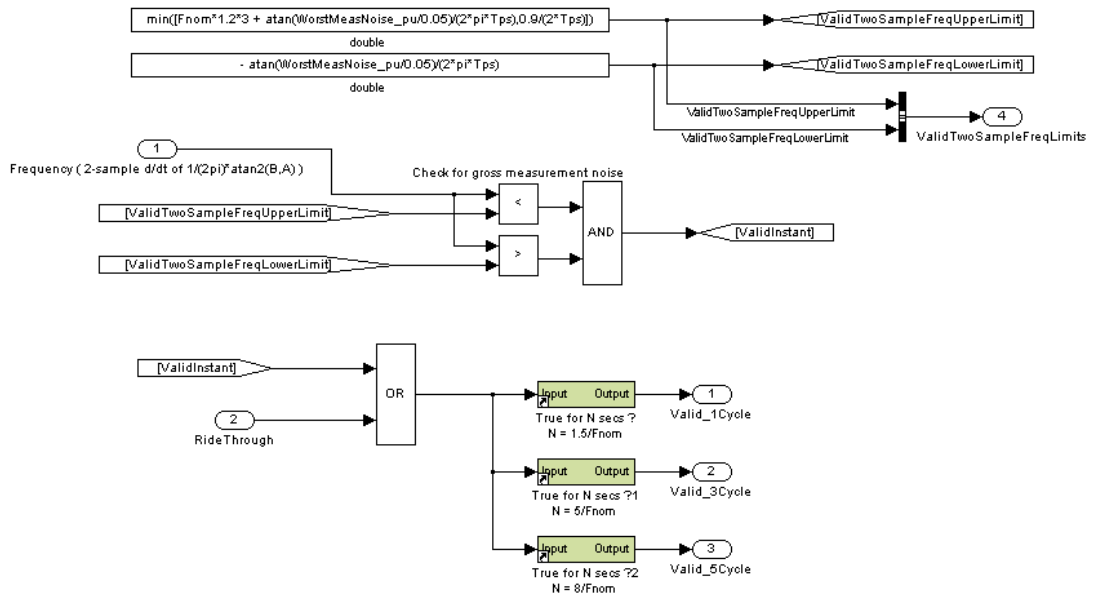
Figure 5-7 : Clarke transform frequency measurement algorithm - detail (3); validity tests

The next stage is to decide whether ride-through should be activated, due to a disturbance; i.e. sudden phase voltage dip (or surge), from a nominal operating condition. Such ride-through action is required as laid down in section 2.7.5. The algorithm for detecting such events is described later in section 5.4.2, since it requires knowledge of the phase magnitudes. The decision process within the Clarke transform frequency measurement is shown in Figure 5-8. If there is such a disturbance, AND the measurement was valid at the "lookback" time, then ride-through can be activated. This will trigger the overall algorithm (Figure 5-5) to hold its output at a constant value, which is taken from the measured frequency value at the "lookback" time. The "lookback" time is set at 1 cycle prior to the detection of the disturbance. This allows for the finite disturbance detection time, and so the frequency used for ride-through is taken from a time which is (hopefully) before any disturbance reached the actual frequency measurement. Ride-

through continues until:-

- a timer expires (set by *MaxRideThroughTime*)

- the phase magnitudes are usable, for a time long enough to make the measurement valid

As soon as either condition is violated, ride-through activity ceases.

Note that a subsequent ride-through action may not be initiated for a fixed period after a ride-through action begins.

Note also that here, and in other certain parts of the algorithms, single-sample delay states (z^{-1}) need to be inserted in some signal paths to avoid "algebraic loops", where these signals are passed back to earlier parts of the algorithm.
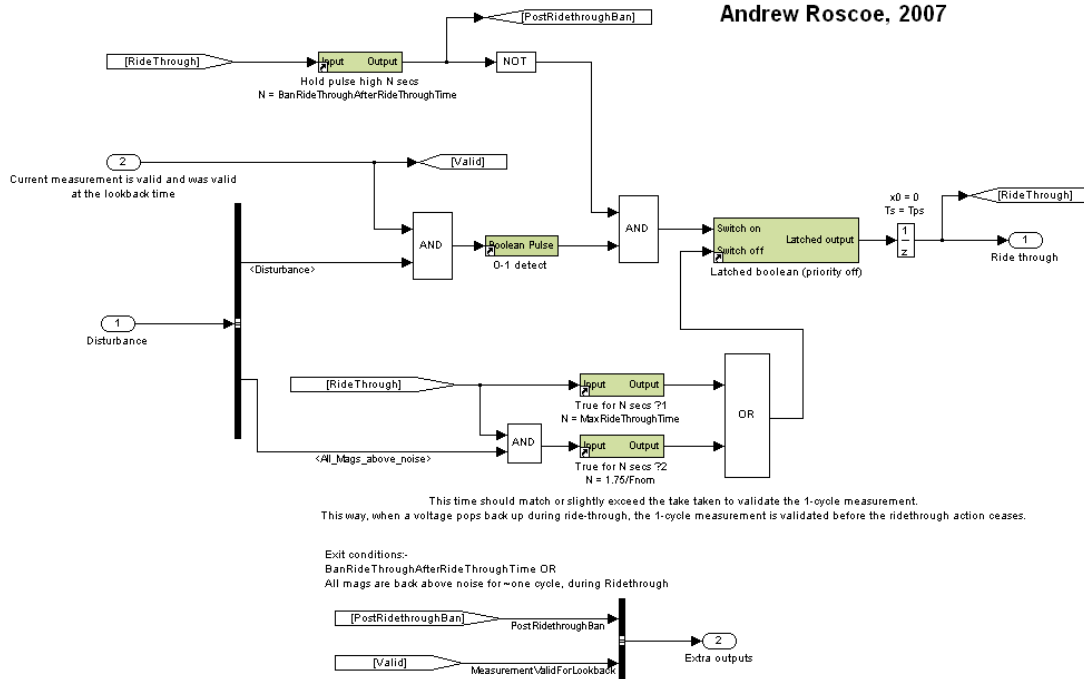
Figure 5-8 : Clarke transform frequency measurement algorithm - detail (4); ride-through decision

The final parts of the algorithm output the primary frequency result and the subsidiary outputs for debug/monitoring/logging (Figure 5-5). A subtle feature is that there are two outputs of frequency. The first is the best estimate, which should always be used to drive any subsequent amplitude/phase measurement blocks. In the case of no valid input signals (or 3-phase instrumentation failure), this output will essentially output noise. The alternate output is set to the nominal frequency if the measurement appears to be invalid.

This alternate output can be used to drive governors containing droop controls, since it will not cause any major primer-mover throttle ramps if there is some form of instrumentation failure. This last point is the reason why the nominal frequency was selected to be output under such conditions, rather than a default to 0Hz.
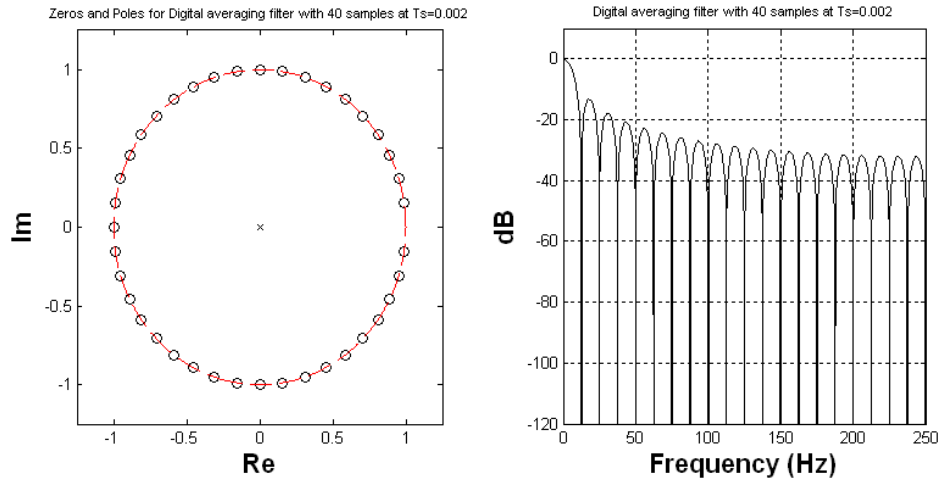
Figure 5-9 : Zeros, poles and Bode plot for a single averaging filter, averaging over 4 cycles
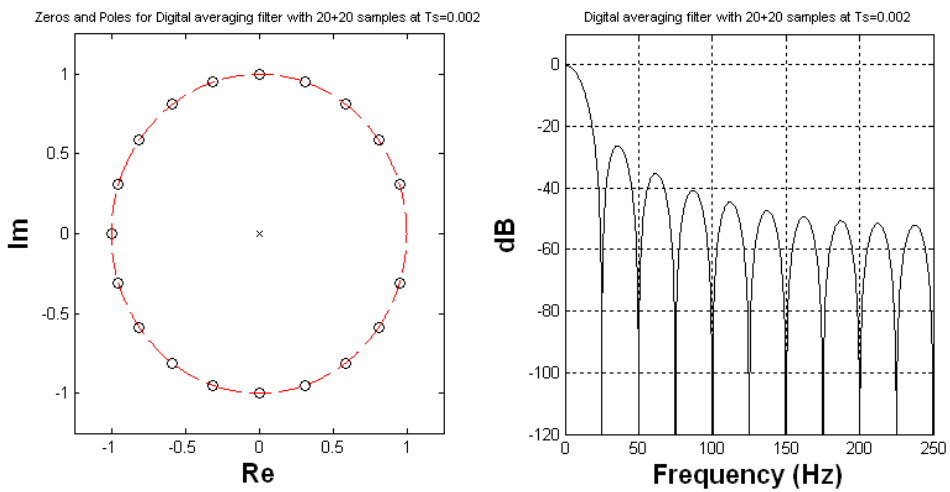
Figure 5-10 : Zeros, poles and Bode plot for a pair of cascaded averaging filters, averaging over 2+2 cycles
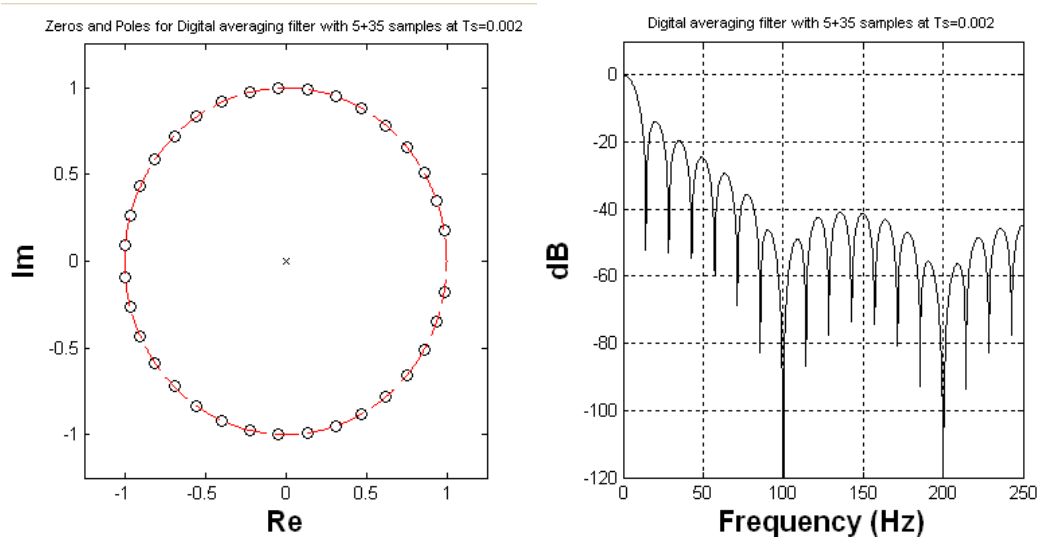
Figure 5-11 : Zeros, poles and Bode plot for a pair of cascaded averaging filters, averaging over ½ +3½ cycles

5.4.2   Method 2 : Clarke-FLL hybrid

The Clarke transform provides a fairly robust method of measuring frequency. However, it does not allow measurement during two-phase faults. To continue measuring frequency during such faults, a system of 3 PLLs with weighted averages to suitably prioritise the outputs of the active or faulted phase(s) could be used. However, as shown in section 5.3, PLLs do not exhibit a desired response where a frequency measurement is required. Instead, this thesis proposes that a FLL (frequency locked loop) is more appropriate. To be precise, a system of 3 single-phase FLLs with weighted average outputs is proposed. Such a set of FLLs aims to track the frequency of the 3 incoming signals, and not the phase. Thus, when signals with a step change in frequency are input, PLLs must exhibit significant overshoot at their frequency output, while the overshoot/ringing response of the FLLs, if correctly designed, is much smaller.

The problem with any PLL or FLL is the initial settling time when a new signal is input. To counter this problem, the FLL can be seeded with a value from another frequency measurement algorithm when appropriate. The seeding can be used to lock the FLL on to the new signal much faster than would otherwise occur. Given that a robust, fast-settling Clarke-transform based measurement has already been designed, the logical solution is to combine a FLL measurement with the Clarke measurement, use the Clarke measurement to seed the FLL when appropriate, and select the best output from either of the two algorithms for subsequent processing. As a bonus, the FLL measurement can be coded to provide not only the frequency measurement, but also the amplitude/phase measurements of the three phase voltages/currents. This reduces the computational burden since the amplitude/phases do not subsequently need to be measured.

228

The total algorithm can be called a "Clarke-FLL hybrid". The first half of the hybrid was presented in section 5.4.1, and can be used as a stand-alone frequency measurement system. The second part of the hybrid (the set of 3 FLLs) is presented in this section. The FLL set can be used without seeding, but is designed to operate using a seed value from the Clarke transform measurement. The FLL algorithm calls upon all the amplitude and phase measuring techniques of chapters 3 & 4. The combination of the two parts forms an extremely effective and robust algorithm. Its operation is described below and summarised on Figure 5-12.

This algorithm contains only FIR (finite impulse response) filters throughout. The exception is that the final measured frequency is fed back to the start of the algorithm, where it is used to help measure the amplitude and phase of each of the 3 input signals. Thus, if the input frequency has a step function disturbance, there is (without the action of seeding), the possibility of some FIR-type settling and ringing response within this algorithm. Note that there is no PI controller, and there is no attempt in this algorithm to track phase. The settling and ringing effects due to the FIR response can be mitigated by use of the seed frequency from the Clarke transform frequency measurement which was described in section 5.4.1. This seeding is extremely effective, because the main algorithm is a frequency locked loop and not a phase locked loop. Thus, seeding the algorithm can be done quickly and effectively. In a PLL this is not the case because seeding it with the correct frequency still requires the PLL to then hunt for the phase before it becomes locked.

The details of the Clarke's frequency measurement have already been described in section 5.4.1. The details of the FLL part of the Clarke-FLL hybrid are presented below in Figure 5-13, Figure 5-14, and Fig. A-1 to Fig. A-8 (see Appendix A). Many of the lower level blocks have been introduced in chapter 3 and require little further explanation. The algorithm is, however, necessarily large and comprehensive. Describing its operation to the lowest level of detail in words is not possible within a single chapter, and would merely repeat the information presented in the figures. The main points of note are summarised below, which, in conjunction with the figures, present the operation of the algorithm to an appropriate level of detail. The performance of this algorithm is demonstrated later in section 5.7.
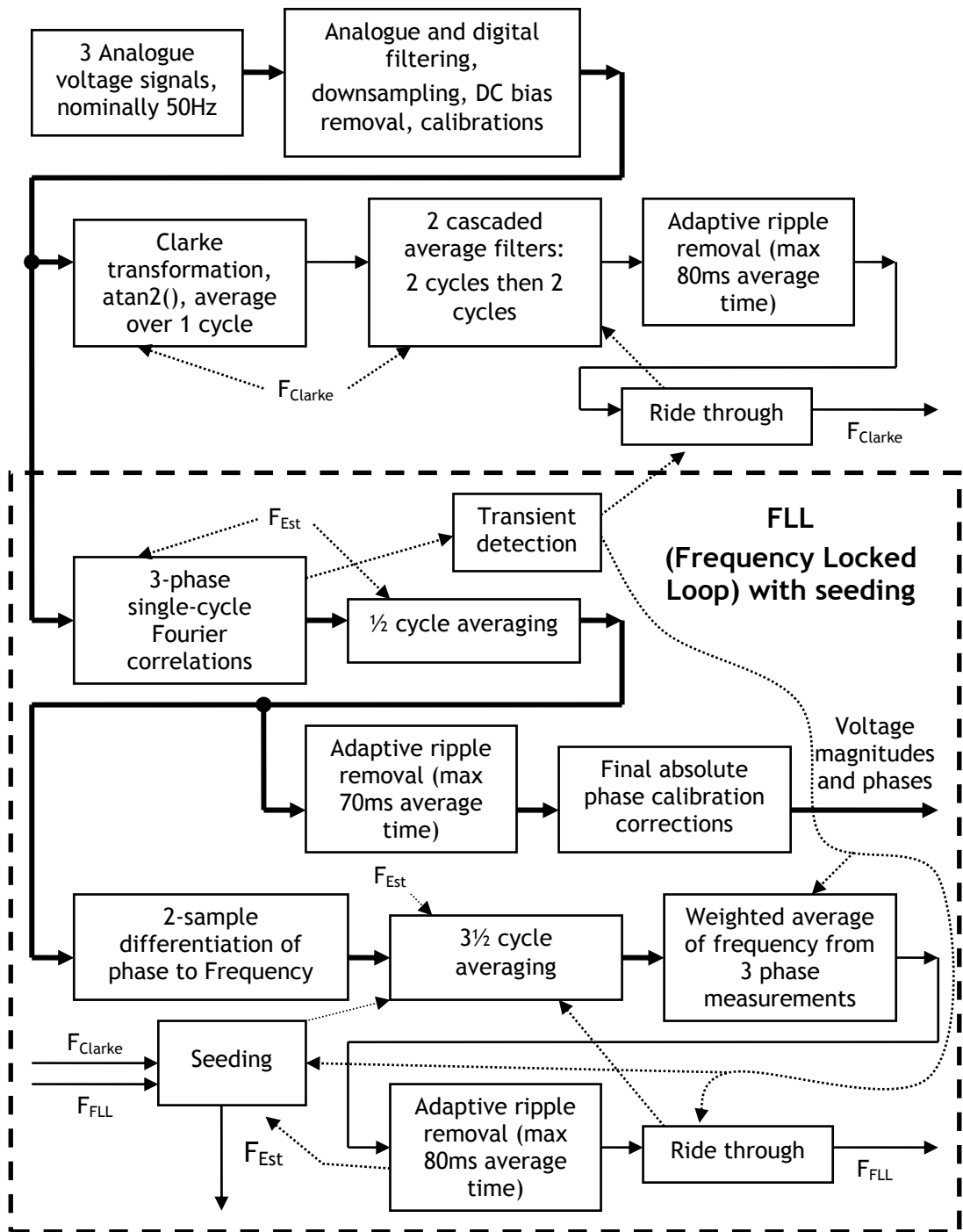
Figure 5-12 : Algorithm summary for Clarke-FLL hybrid

Figure 5-13 : FLL – overview

Measure frequency, fundamental amplitude, approximate all-harmonic RMS and approximate THD.
Three phase

Andrew Roscoe, 2007
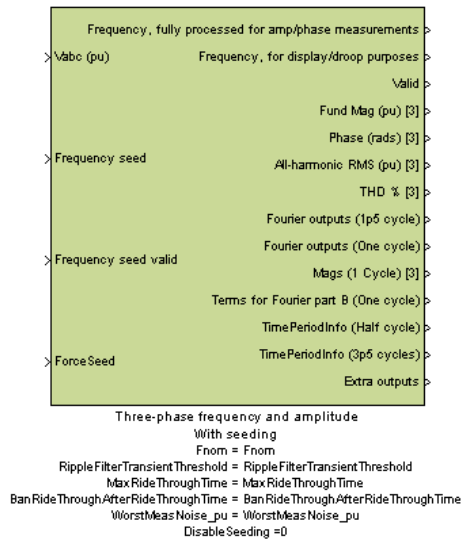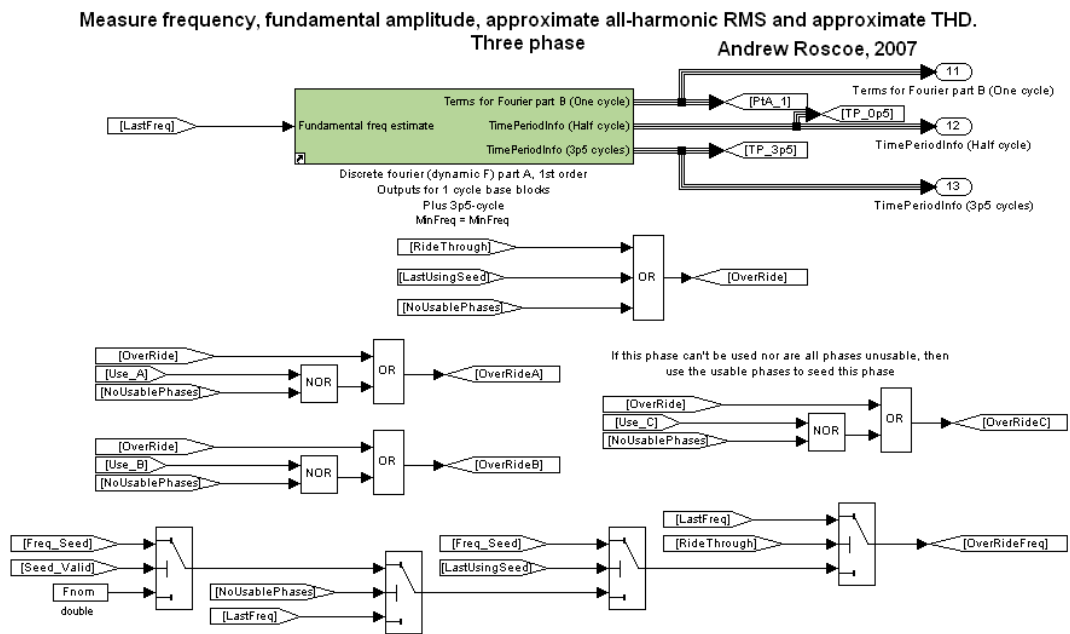
Figure 5-14 : FLL – detail [this and next two pages]
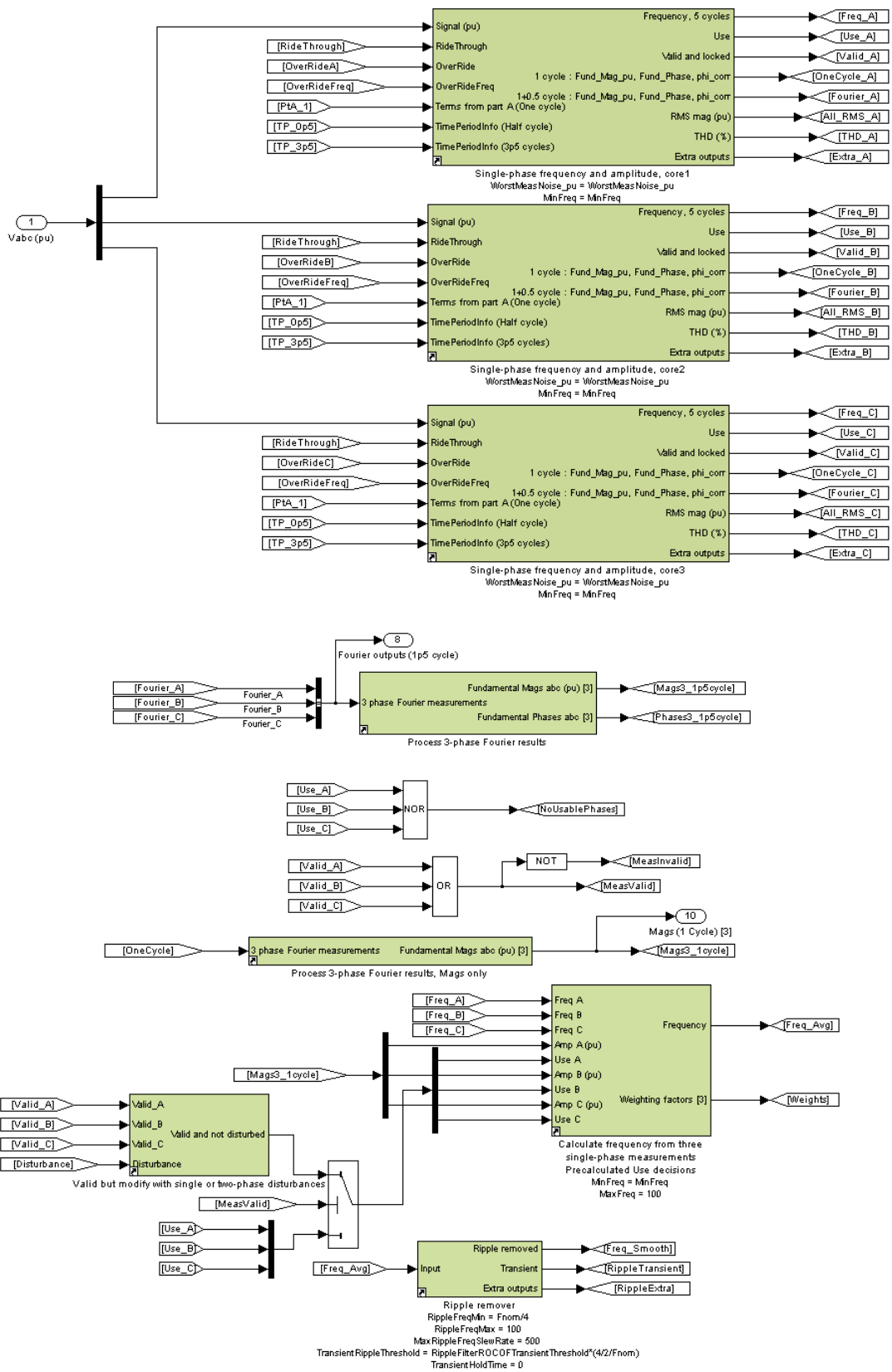
Figure 5-14 : FLL - detail [previous, this and next pages]

232
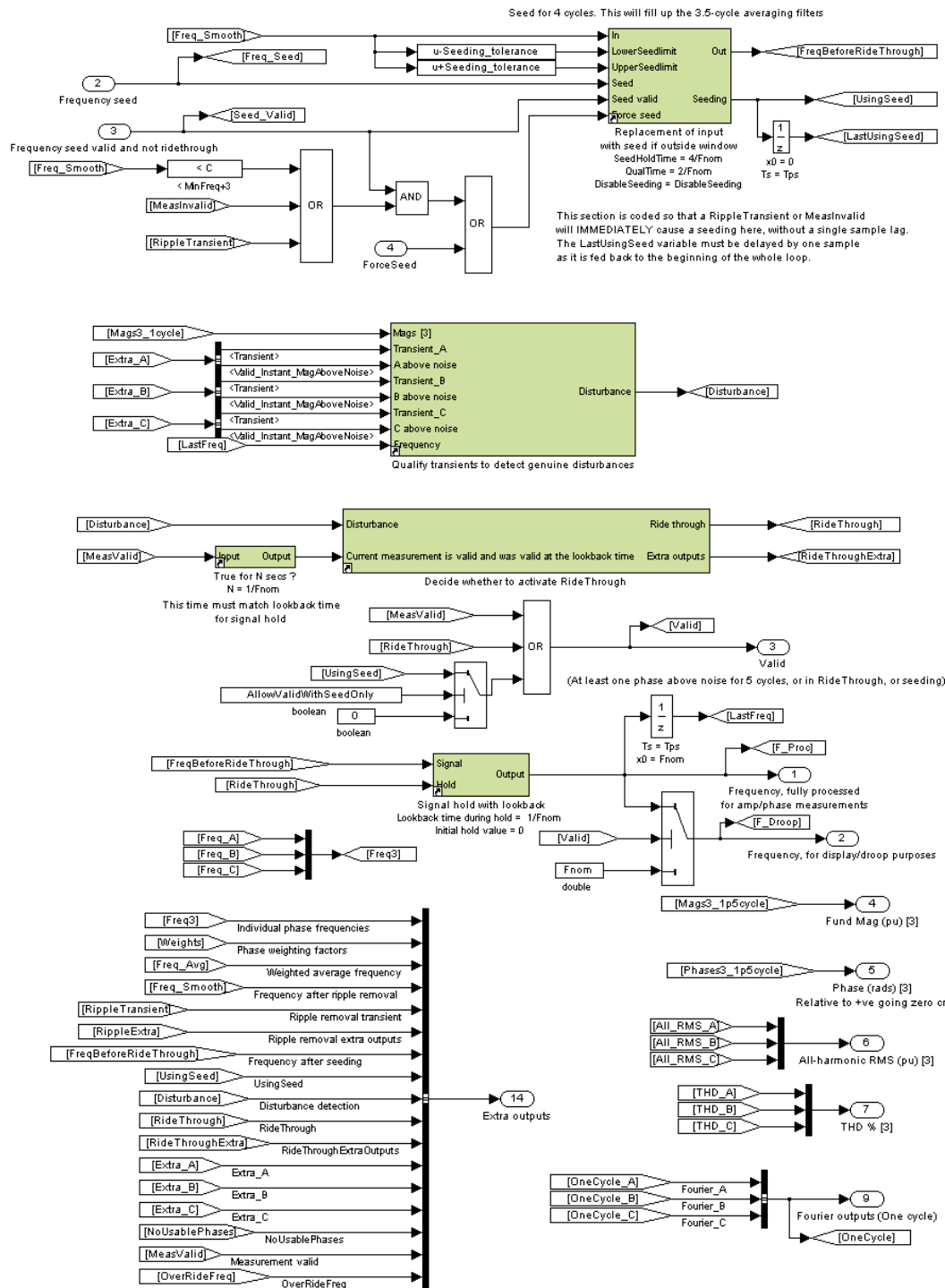
Figure 5-14 : FLL – detail [this and previous 2 pages]

The core of the algorithm is a set of 3 Fourier analysis blocks (Fig. A-1). These are "1(NC)+0.5" (see section 3.13) blocks using 1st order interpolation/integration for the 1-cycle and ½-cycle averaging blocks. This stage is just as described as in section 3.9. The outputs of the 3 blocks are the amplitudes and phases of the fundamental waveforms. These can be used directly for voltage measurement purposes, synchronisation checking

etc., and hence there is good optimisation of CPU use by making these blocks the core of the Clarke-FLL hybrid algorithm. All of the other frequency measurement methods presented required subsequent additional amplitude/phase measurement blocks to be executed in order to determine the amplitude (and disturbance) information for each phase.

The frequency of each phase is measured within the core blocks by differentiating the absolute phase (relative to a positive-going zero crossing) of the waveform on each phase, at the output of the "1(NC)+0.5" block. Then, as in the Clarke transform frequency measurement, an additional 3½ cycles averaging can always be applied to realise a measurement with ~5 cycle latency. In the case of the Clarke transform measurement, a 1+2+2 cycle averaging scheme was used which gives the best noise rejection (see section 5.4.1). Within the Clarke-FLL hybrid, a 1½-cycle measurement has already been carried out within the "1(NC)+0.5" block. A compromise between noise rejection and CPU loading is then to apply an additional 3½-cycle averaging filter to make up the averaging to 5 cycles, using a 1+½+3½ cycle measurement system. The decrease in noise rejection relative to the 1+2+2 cycle measurement system is relatively small, as shown by Figure 5-10 & Figure 5-11.

During certain circumstances, the 3½-cycle filters are filled (pre-loaded) with values other than the actual measured values. These circumstances are:-

- During fault ride-through

- When the seed frequency is being used

- When the phase voltage appears to be very low (in the noise), and at least one of the other phases appear to have a usable signal voltage present

This pre-load action helps to speed up settling of the averaging filters when a genuine signal (re-)appears.

When a voltage level above the level of noise appears on a phase, a sequence of flags gives an indication of the validity of the frequency measurement from that phase:-

- When the voltage has been significant for 2 cycles (40ms), the frequency measurement begins to be used within the algorithm as a whole. This is described by the flag "Use" and "Use_A", "Use_B" etc.. If this flag is low, the measurement of frequency from the phase is discarded, unless all phases have low voltages in which case all phase measurements are used in equal weights.

- When the voltage has been significant for 5 cycles (100ms), the frequency

measurement can be valid, but it must also be locked to be declared fully valid

- Lock, for each phase, is determined by analysis of the rate of change of phase of the input signal, relative to the correlating Fourier waveform at the measured frequency. This quantity will drop towards zero when the algorithm is at steady state with a steady frequency input. Lock is determined by analysing this signal through a low-pass filter and an "abs" function (see Fig. A-1). This provides the most reliable way of determining lock and unlock without using more CPU-intensive functions such as rolling buffers, and without incurring spurious unlock signals during transient voltage spikes.

- When any of the three phase measurements is valid (and locked), then the measurement as a whole is valid.

The logic in the above bullet points is important to get right, because being too conservative results in an algorithm which can never get started (it gets stuck in an "invalid" state), and being too lax allows incorrect answers to appear at the output with a "valid" status.

The answers from the 3 core frequency measurement blocks are combined together within a weighted averaging algorithm, shown in Fig. A-3 and Fig. A-4. This uses weightings determined from the phase magnitudes, applying higher weightings for signals with nominal 1pu magnitudes, and lower weightings for signals which are higher or lower than 1pu. In addition, logic of Figure 5-14 can set the weightings to 10^{-9} if the frequency measurements from each core block are invalid or unlocked, or if a single or two-phase disturbance is in progress (Fig. A-7). This allows clean, sustained ride-through of single and two-phase disturbances. If none of the 3 core blocks is valid or locked, then the logic does allow measurements to be used if the phase voltage magnitude has been significant for 2 cycles and the "Use_X" flag is set for that phase. This allows the algorithm to get going from a "cold start", even in then absence of a usable frequency seed.

A ripple filter is applied, as for the Clarke transform frequency measurement. Again, the maximum latency allowed is 4 cycles, for the reasons discussed in section 5.4.1. The ripple filter switches itself off during rapid frequency changes to keep the latency within specification.

The seeding algorithm is spread between Figure 5-14 and Fig. A-5. Seeding will occur if the Clarke transform frequency measurement is valid AND any of the following conditions are true:-

- The local frequency measurement is more than 1Hz different to the Clarke transform measurement

- The local frequency measurement is close to or less than the variable *MinFreq*, which sets the lowest frequency which can accurately be measured inside the Fourier measurement blocks due to the length of the averaging buffers. These must be predefined at compile-time as buffers with approximately X/(*MinFreq**Ts) samples stored in memory, where X is the number of cycles of averaging for each buffer. When the actual frequency drops below this value, the core Fourier blocks tend to unlock due to the ripple which appears. Note, this problem does not occur with the Clarke transform frequency measurement as the phase trajectory is a straight line, so an inaccurate averaging period does not produce such a rippling effect unless unbalance is large.

- If the ripple filter reports a transient event, seeding is a sensible option because it is likely that the Clarke transform frequency measurement will produce a better result due to its very fast settling.

To trigger ride-through, disturbances in the fundamental magnitudes of the three phase voltages must be detected quickly, and compared against thresholds of tolerance. This is done using the algorithm of Fig. A-7. The inputs to this algorithm are outputs from the 1-cycle latency Fourier correlations, namely the fundamental voltage magnitudes, and "transient" flags for each phase which are determined within the Fourier correlation block as shown in Figure 3-42. A valid disturbance is flagged when the voltages and frequency start at values close to nominal, and then at least one of the phase magnitudes deviates rapidly. This rapid deviation can occur due to a fault, disconnection, or surge transient. In the cycles that follow, the disturbed phase voltage may be immeasurably low, unsinusoidal, or contain a voltage spike. Any of these, or even a large step change in phase voltage, can cause inaccuracy in the frequency measurement taken from the disturbed phase. The disturbance flags generated by this algorithm will usually be within ≤½ a cycle of the event occurring, due to the settling characteristics of the 1-cycle Fourier correlations.

The next step is to decide whether a disturbance flag should trigger a ride-through action. This is done using the algorithm of Fig. A-6, which is very similar to the equivalent decision process for the Clarke transform frequency measurement algorithm shown in Figure 5-8. The difference in this case is that because the measurement can function during two-phase faults, the algorithm will exit from the ride-through condition if any of the three phase voltages re-appear to usable levels. The Clarke transform measurement equivalent

236

requires all three phase voltages to re-appear to usable levels to cause this same action.

The overall validity of the measurement is determined by any one of the following conditions being true:-

- The local measurement is valid

- Ride-through is active

- The seed is being used (this can only happen when the seed itself is valid)

The final step is to implement the ride-through, if it is active. This is done as shown in Fig. A-6 and Fig. A-8. When a ride-through action begins, the frequency is sampled and then held at a steady value for the duration of the ride-through state. The trick is to take the sample from a time prior to the beginning of the ride-through state. Here, a 1-cycle look-back is used. This time must be set to slightly longer than the disturbance detection might take, which might be ½ a cycle inside the algorithm plus the latency of the analogue & sampling hardware. This avoids taking the sampled value from a time after the disturbance began, when the frequency measurements may be already distorted.

This leads to the final output frequency. As for the Clarke transform measurement, there are two output ports. The first is unmodified, and should be used to feed any other algorithms which would prefer a "best estimate" answer even if the measurement is not valid. The second pin is for display/logging/droop actions, and reverts to an output at f_{nom} (nominal frequency) when the measurement is invalid. This avoids "random" frequencies appearing on displays etc. when no voltages are present and the instrumentation noise is the dominant signal. It also minimises the risk of droop controllers railing during a momentary or complete instrumentation failure.

5.4.3 Method 3 : Fixed-reference Fourier frequency measurement with seeding

This technique is in many ways similar to the FLL of section 5.4.2. In this case, however, the IIR settling response is eliminated by using a fixed 50Hz (f_{nom}) reference for the Fourier correlation waveforms. This means that the correlations do not make good representations of the actual magnitude and phase of the input signals, as for any frequencies off-nominal the ripple at the output of the 1-cycle correlation is large. However, when averaged using cascaded exact-time averaging filters this ripple can be mostly removed, by the processes described in section 4.1. Thus, a reasonable estimate of frequency can be made, on a phase-by-phase basis, which allows two-phase faults to be tolerated. As with the Clarke-FLL hybrid, the possibility exists of errors and locking to sub-harmonics/harmonics

if the input signal is sufficiently far below or above nominal. For this reason, and to benefit from the faster settling available from the Clarke transform measurement, this algorithm can elect to use the Clarke transform measurement as a seed when it makes sense. A drawback of this algorithm is that the amplitudes/phases must be subsequently re-measured using separate measurements, to avoid the small DC bias error at 0Hz which appears as described in section 4.1.

The coding details are presented in appendix E.1.

5.4.4   Method 4 : Three customised single-phase PLLs with weighted averaging and seeding

In order to compare the performance of a phase-locked-loop (PLL) with the above algorithms, a PLL-based solution has been coded, using the same concepts of fault ride-through and seeding. The code is shown in appendix E.2. It was chosen to implement a set of 3 single-phase PLLs rather than a single 3-phase dq-based PLL, because this solution allows operation during two-phase faults whereas the 3-phase PLL is vulnerable in this scenario. There is similarity to the previous two methods described, but it should be noted that some of the seeding details are different. This is because the single-phase PLLs contain an internal seeding algorithm. This is carefully coded because the seeding can only initialise the frequency part of the PLL; i.e. the PI controller. Once this is done, the seeding must be released and after this time the PLL must still hunt to find the correct phase lock. Thus, the seeding is more difficult to arrange, and less effective, than for the previous two algorithms.

5.4.5   Method 5 : Zero crossings

For comparison, the ride-through techniques and weighted averaging algorithms were also applied to a zero-crossings based technique. This would not be expected to perform so well, as only the pairs of samples nearest the zero crossings are used for the measurement, and thus the instrumentation noise is not so well attenuated. Also, subsequent exact-time averaging blocks do not attenuate ripple due to harmonics so well, as the resulting ripple frequency is not at exact multiples of the fundamental frequency. The algorithm requires no seeding as there is no FIR-type feedback loop. The primary measurements are made using the previous 5 cycles (10 zero crossings). The algorithm is robust against single and two-phase faults, as presents quite a low computational burden (see section 5.6). The code is presented in appendix E.3.

5.4.6   Method 6 : Three Jovcic's PLLs with weighted averaging

For comparison, a system of 3 PLLs to the Jovcic (2003) design is also implemented. The outputs of these are combined using a weighted averaging (as described in section 5.4.2) to optimise the performance during single and two-phase faults. The reader should refer to the original reference for the detailed PLL implementation details. The only differences between the implementation used in this thesis and the original Jovic code is that the 2nd harmonic cancellation algorithm has been simplified (see section 3.8) and a few checks have been added to avoid divisions by zero and overflow errors. There is no seeding and no detailed ride-through code added to this algorithm. It should be noted that Jovcic's PLL uses a low-pass filter with a cut-off frequency of only 1.3 Hz (ω=1/0.12=8.33 rad/s), which has a "5RC" settling time of about 0.6 seconds. This smoothes the output dramatically and causes substantial measurement latency.

5.4.7   Method 7 : SimPowerSystems 3-phase PLL

The SimPowerSystems 3-phase PLL is used without modification, for comparison with the other solutions. It should be noted that this (and the SimPowerSystems single-phase PLL), use a rate limiter set to ±12Hz/s combined with a 2nd-order low-pass filter with a cut-off frequency set to 25Hz, at the frequency output port. These two filters cause a large measurement latency during initial locking to a new signal at a new frequency. The algorithms are small compared to the proposed Clarke-FLL hybrid, but since the measurement of 3 phase amplitudes requires code additional to the SimPowerSystems PLL, the execution speed advantage over the Clarke-FLL hybrid is not great (see section 5.6).

5.4.8   Method 8 : Three SimPowerSystems single-phase PLLs with weighted averaging

Three SimPowerSystems 1-phase PLLs are used without modification. The outputs from the three PLLs are combined with a weighted average system to optimise the performance during single and two-phase faults. Again, the algorithms are small compared to the proposed Clarke-FLL hybrid, but since the measurement of 3 phase amplitudes requires code additional to the SimPowerSystems PLLs, the execution speed advantage over the Clarke-FLL hybrid is not great (see section 5.6).

5.5 Summary of considered techniques for frequency measurement

Table 5-1 shows a brief summary which compares the properties of the 8 trialled algorithms.

Algorithm	Three-phase measurement (3P) or 3x single-phase measurements using weighted average (3x1P W)	Robust during two-phase faults	Robust for wide frequency range? With (and without) seeding	Includes Ride-through coding
Clarke transform	3P	X	✓	✓
Clarke-FLL hybrid	3x1P W	✓	✓ (X)	✓
Fixed-ref Fourier with seeding	3x1P W	✓	✓ (X)	✓
3x 1-phase PLLs with seeding	3x1P W	✓	✓ (X)	✓
Zero crossings	3x1P W	✓	✓	✓
3x Jovcic's PLLs with weighting	3x1P W	✓	X	X
3x SPS PLLs with weighting	3x1P W	✓	X	X
SPS 3-phase PLL	3P	X	X	X

Table 5-1 : Summary of considered frequency measurement techniques

5.6 Benchmarking analysis

The 8 candidate methods for frequency measurement vary both in algorithmic complexity and also in the requirement for "hard" maths functions. The "hard" maths functions are those such as trigonometric functions and square root operators, as defined in section 3.13.1, Table 3-3. Such functions require significantly more time to execute than additions/multiplications or logical decisions (see Appendix G). An important consideration in the design of the Clarke-FLL hybrid measurement is that its overall computational burden and execution time should be competitive with alternative solutions. This is shown below. An important aspect of this analysis is that the Clarke-FLL hybrid algorithm provides measurements of both frequency and the amplitudes/phases of the 3 voltages/currents, in a carefully integrated manner to minimise CPU loading. The other 7 candidate frequency measurement algorithms require additional code to provide (in particular) accurate amplitude measurements. When the effect of this extra required code is accounted for, the total CPU burden of the Clarke-FLL hybrid is found to be competitive with that of the other algorithms.

The number of "hard" maths functions (see section 3.13.1 Table 3-3 for the definition of

"hard" functions) and delay buffer blocks (for the exact-time averaging) can be counted up for each algorithm combination, to provide a simple measure of CPU loading for each algorithm. These counts provide a reasonable but not comprehensive picture of CPU loading, since the many other more trivial but numerous calculations and operations will also contribute to the CPU load. Figure 5-15 below shows these counts for the 8 candidate algorithms, when used to provide a measurement of frequency and the fundamental amplitude/phase components of voltage, for a 3-phase voltage waveform set. The code to measure all-harmonic RMS and THD is not included in this analysis, since RMS and THD have been shown to be inaccurate measures at the low sample rates addressed in this thesis.

Number of "hard" maths functions (sin/cos/atan2/sqrt) for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase
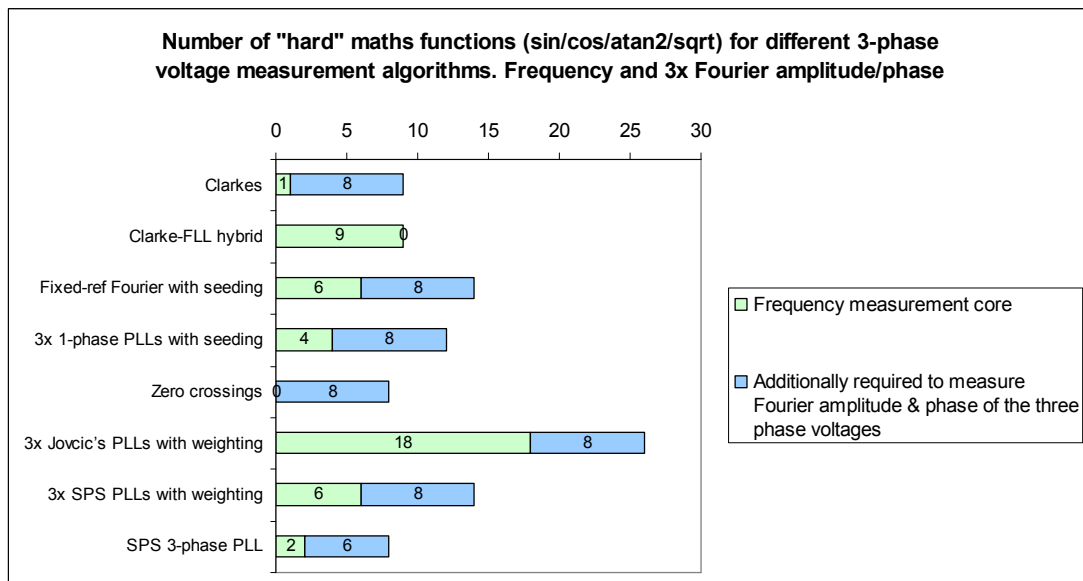
Figure 5-15 : Number of "hard" maths functions (sin/cos/atan2/sqrt) for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

The Clarke-FLL hybrid compares well with the other candidate algorithms in Figure 5-15. The Zero crossings and SimPowerSystems 3-phase PLL algorithms require fewer "hard" functions.

The Clarke-FLL hybrid contains a lot of delay buffers (69), compared to most of the other candidate algorithms (see Figure 5-16)[1]. In particular, the zero crossings and

---

[1] 3 delay buffer blocks are required for each signal averaging stage. Thus 36 buffer blocks are required to implement the "1(NC)+0.5" process on a set of 3 voltages. (3 blocks per average, 2 averages per phase for path A/B or Mag/Phase, 2 stages, 3 phases; 3*2*2*3=36).

SimPowerSystems PLL algorithms require fewer delay blocks, since their signal processing mechanisms are simpler.
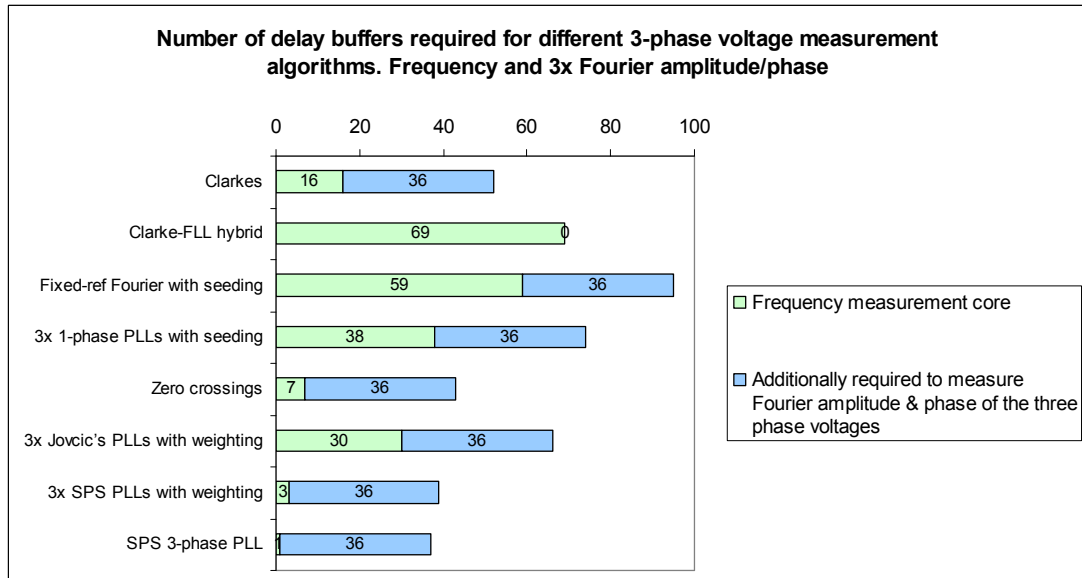
Figure 5-16 : Number of delay buffers required for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

Benchmarking experiments on the Infineon TC1796 microcontroller reveal that the "hard" math functions take about 0.75μs to execute, while the delay blocks take about 0.25μs (see Appendix G). Based upon these timings and the numbers of "hard" maths functions and delay blocks, optimistic lower limits on the possible execution times for each algorithm can be simply estimated. These are shown in Figure 5-17. The Zero crossings and SimPowerSystems PLLs will have relatively low execution times due to their relative simplicity and the lack of any seeding functionality. Other algorithms will take longer. The Clarke-FLL hybrid compares reasonably (only 57% higher than the quickest solution, the SPS 3-phase PLL), and the prediction from this analysis is that the execution time should be somewhere in the excess of 24μs. A real benchmarking of the final algorithm is carried out in Appendix G, and the actual execution time for this algorithm is measured at 68μs.
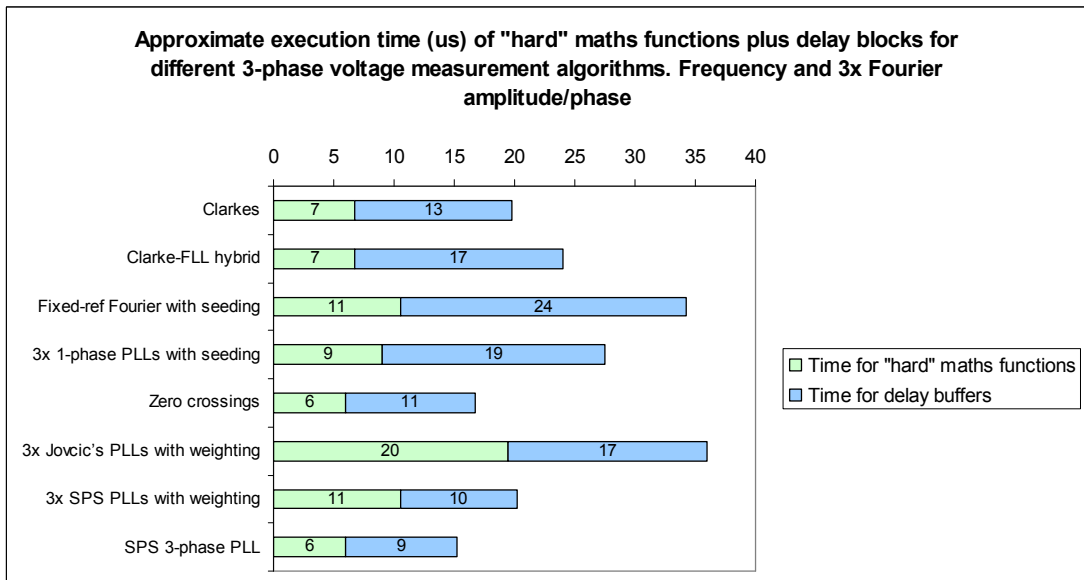
Figure 5-17 : Approximate execution time (us) of "hard" maths functions plus delay blocks for different 3-phase voltage measurement algorithms. Frequency and 3x Fourier amplitude/phase

Thus, the "hard" maths functions and delay blocks account for 35% of the overall execution time. The delay blocks account for about 17/68*100 = 25% of the overall execution time. As found in the benchmarking experiments in Appendix G, the execution time of the delay blocks appears to be limited by memory addressing and access time. This will be highly dependent upon the hardware platform used, and may be significantly faster (or slower) for other processors. On the TC1796, with an execution time of 68μs, at a frame rate of 10 frames per cycle (500 samples per second, a 2000μs frame rate), this leaves about 1932μs, 97% of the frame available for other tasks. These include analysis of 3-phase currents, positive/negative sequence analysis, analysis of further power system nodes, relaying actions, control algorithms, data logging, communication, etc. Some execution times for larger frequency/voltage/current and sequence/power-flow analysis algorithms have been measured and are shown in section Appendix G. In particular, the entire algorithm required to perform a full nodal measurement takes 156μs, less than 8% of the 2000μs frame time. This algorithm measures frequency, voltages and currents, with full sequence analysis and power flow calculation. All the amplitude and phase calibration functions are accounted for, including de-skewing of multiplexed ADC channels.

5.7   Comparison of 8 candidate frequency measurement algorithms

To compare the performance of the 8 candidate algorithms, results are presented here for the test input "waveform 1" specified in Table 2-16, against the specifications of Table 2-14. To verify the performance against specification, lower and upper test-line limits were generated, within which the measured signal should fall. The lower limit is defined by the following process:-

- Evaluate the instantaneous lower limit, as the actual frequency minus the allowed specification at any time. Generally the allowed specification is ±0.005 Hz (±0.0001 pu), but gets larger during faults and dips. Note that fault ride-through specifications mean that the specification does not widen until at least the specified ride-through time has elapse after the inception of the fault. In these simulations, this time was set to 2 cycles (40ms), but this is a configurable parameter for all the algorithms with fault ride-through code.

- Evaluate the lower limit by taking the minimum of all the instantaneous lower limits over the previous 5 cycles (100ms), since this is the required response time of the measurement as defined in Table 2-14.

- The upper limit is defined in a symmetric fashion compared to the lower limit.

The test waveform was synthesised at 1000 Sa/cycle (50 kSa/s) and the hardware low-pass filter was simulated at this same data rate, so as to be sure of accurately representing the attenuation of the higher-order harmonics (see appendix B.2.2).

The instrumentation noise was set to 0.005pu RMS (46dB SNR), with 2 bits RMS additional quantisation noise in the ADC, as specified by Table 2-15. The ADC and 6-tap FIR pre-filter of section 4.5 was implemented at 3000 Sa/s, a 6x over-sampling relative to the main algorithms, for the same reasons discussed in that section.

The main algorithms all execute at 500 Sa/s, 10 samples-per-cycle at nominal 50 Hz (t_s=1/500 s). All algorithms can operate at this low sample rate, although the Jovcic's PLL has been seen to be vulnerable to harmonics at this rate. Jovcic's PLL appears to be much more stable at 2 kSa/s or above.

At all points during the 60-second simulation, the result from each candidate algorithm is compared to the specification lines. An error score is allocated for each candidate algorithm if the algorithm result falls outside the allowed specifications. The error score at

each time step (of length t_s) is set to:-

$$Error = 0 \{ f_{Measured} > f_{LowerLimit} \} and \{ f_{Measured} < f_{UpperLimit} \}$$

otherwise

$$Error = \left(\frac{t_s}{60} \right) \left[\frac{(f_{Measured} - f_{Actual})}{(f_{UpperLimit} - f_{Actual})} - 1 \right] \{ f_{Measured} > f_{Actual} \}$$

otherwise

$$Error = \left(\frac{t_s}{60} \right) \left[\frac{(f_{Actual} - f_{Measured})}{(f_{Actual} - f_{LowerLimit})} - 1 \right] \{ f_{Measured} < f_{Actual} \}$$

(5.5)

This error scoring system allocates error scores proportionally to the measurement error divided by the specification allowance. Thus, once the specification has been violated, the error score increases slowly for measurement only just outside the specification, but much more quickly for large violations.

The factor of $t_s/60$ in (5.5) scales the errors to an average figure over the 60 second test duration. The error at each time step is then clipped to a maximum value of 100. The cumulative error is taken by adding all the resulting error scores at each time step. Thus, the cumulative error represents the average amount by which the measurement is outside the specification window, on a proportionate basis to the window size. This provides a fair scoring system for comparison of the algorithm performances.

5.7.1   Results overview

It has been found possible to achieve the tightest target specification of ±0.005 Hz in all but a few transient events and at the worst frequencies, by using method 2, the Clarke-FLL hybrid. This is the algorithm of choice, both by the results of Table 5-2 and Table 5-1 . Even during the transient events and at the worst frequencies, the ±0.005 Hz specification is only narrowly missed, and a ±0.01 Hz error specification is almost always achieved.

The overall cumulative error scores for the 8 candidate methods tested, using a basic specification of ±0.005 Hz, and the Waveform 1 and 2 test inputs (Table 2-16 and section 2.11.2) are:-

Algorithm	Cumulative error scores, for waveform 1 test input (see Table 2-16) (average amount by which the measurement is outside the specification window, on a proportionate basis to the window size)	Cumulative error scores, for waveform 2 test input (see Table 2-17), steady state between 44 & 55 Hz (average amount by which the measurement is outside the specification window, on a proportionate basis to the window size)
Clarke transform	0.3343	0.00613
Clarke-FLL hybrid	**0.0094** (2.785 if seeding is disabled)	**0.000877**
Fixed-ref Fourier with seeding	0.0331	0.0124
3x 1-phase PLLs with seeding	0.0872	0.0819
Zero crossings	1.657	0.1876
3x Jovcic's PLLs with weighting	21.31	8.566
3x SPS PLLs with weighting	23.57	3.794
SPS 3-phase PLL	23.68	3.819

Table 5-2 : Summary of cumulative error scores for 8 candidate frequency measurement algorithms, using test Waveforms 1 (Table 2-16) and 2 (Table 2-17)

The results for the waveform 1 test input scenario are investigated in detail in the following sections. Figure 5-18 shows an overview of the frequency profile for the waveform 1 test profile, together with the results for the Clarke-FLL hybrid. The traces are identified as follows:-

- Black solid line : Measurement

- Red dash-dot : Actual synthesised signal frequency

- Blue dashes : Lower and Upper limit lines
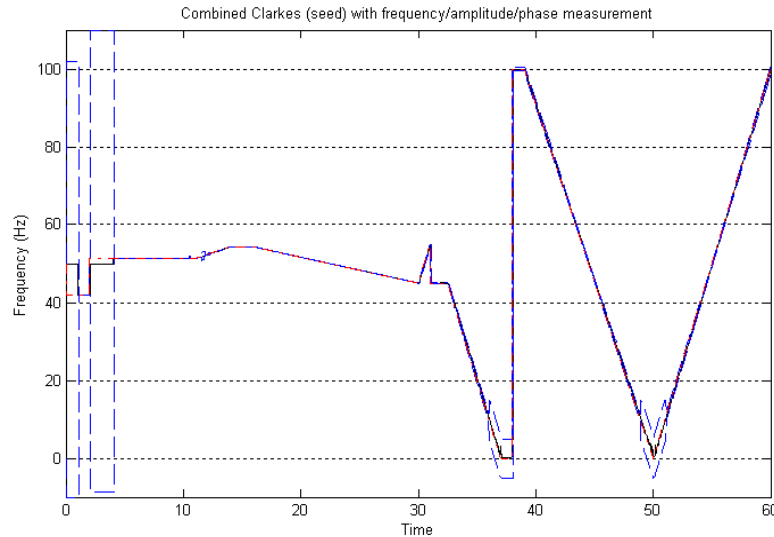
Figure 5-18 : Overview of frequency profile, limit lines & results for waveform 1 (Table 2-16)

5.7.2 Results in detail; initial settling

The simulation deliberately includes an initial period of three-phase signal at 42Hz, balanced 1pu voltage magnitudes between t=1 and t=2s. This has the effect of disturbing all algorithms from any preset values of nominal 50Hz. The first really interesting part of the simulation is thus the time at t=4s, when the signal at 51.282Hz is input. Figure 5-19 shows the results.

On all the graphs in subsections which follow, the traces are identified as follows:-

- Black solid line : Measurement

- Red dash-dot : Actual synthesised signal frequency

- Blue dashes : Lower and Upper limit lines

On Figure 5-19 and the following pages, the results of the Clarke-FLL hybrid algorithm (the proposed algorithm) is shown on the top-right plot, labelled "Combined Clarkes (seed) with frequency/amplitude/phase measurement"

On Figure 5-19, the plots for the 2 SimPowerSystems PLLs and the Jovcic's PLLs need to be displayed on different scales as their settling is much, much slower than the other 5 candidate methods. The seeded methods all settle quickly and accurately, as they use the Clarke transform seed while the main algorithm is settling. The seeded 3x single-phase PLLs show a secondary undesirable transient at t=4.25 seconds, when the seeding turns off and the PLLs complete their locking to the signal phases. This error could be reduced by using the Clarke transform measurement for longer, but this defeats the purpose of using the PLLs at all and emphasises that their worth as a plain frequency measurement device is limited by this phase-locking requirement.

The Clarke-FLL hybrid algorithm (and the other measurements which use the Clarke's frequency measurement seed) all settle to approximately the right value very quickly, within 40ms. This is due to the initial use of just a single full cycle of data. Then, over the next 5 cycles, the Clarke's seed measurement gets more accurate as the 3-cycle and 5-cycle averaging is progressively engaged (see section 5.4.1 and Figure 5-5, Figure 5-6).
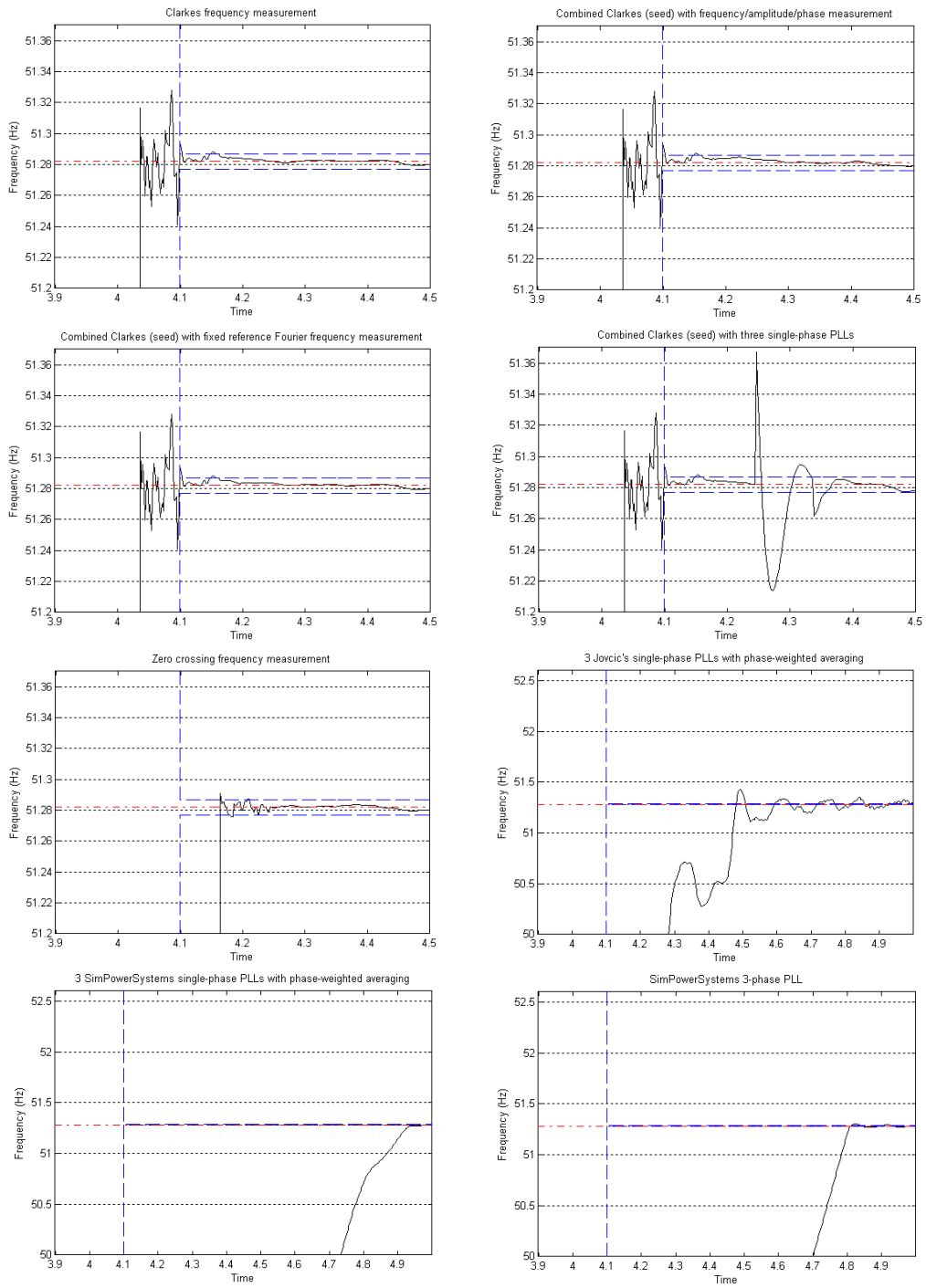
Figure 5-19 : 8 candidate methods; initial settling to 51.282 Hz

5.7.3 Results in detail; increasing influence qualities at 51.282 Hz
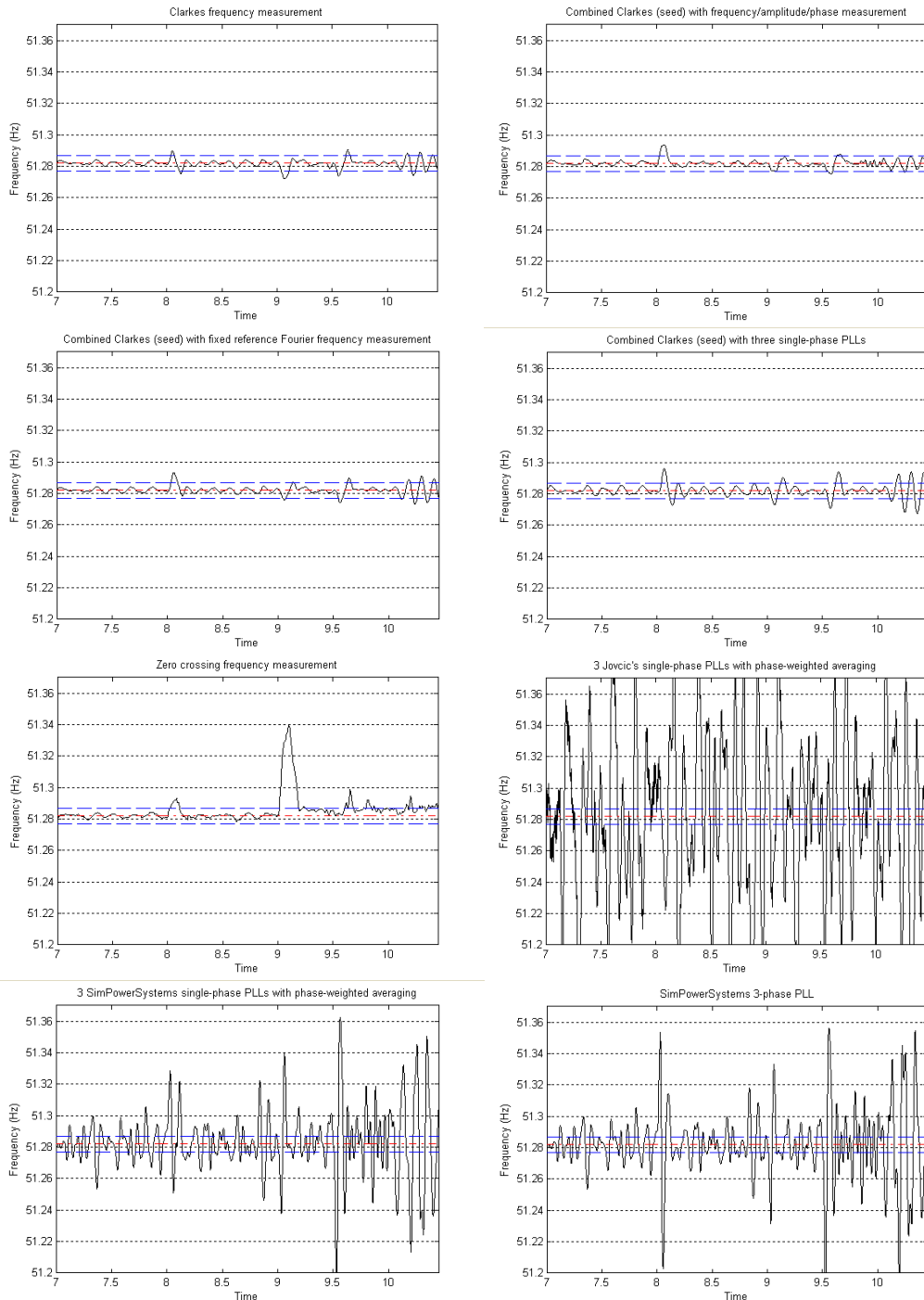
Figure 5-20 : 8 candidate methods; increasing influence qualities at 51.282 Hz

In these plots, 10% unbalance & 2% zero sequence is introduced at t=8s, harmonics (53% THD$_V$) is introduced at t=9s, and inter-harmonics are introduced at t=9.5s. 8% step flicker @ 13.5Hz is introduced at t=10s. The Clarke-FLL hybrid gives results which are not bettered by any other methods, but equalled by 3 others developed during this thesis. The zero crossing algorithm, Jovcic's PLL and SimPowerSystems PLLs give unacceptable results.

5.7.4   Results in detail; phase jump
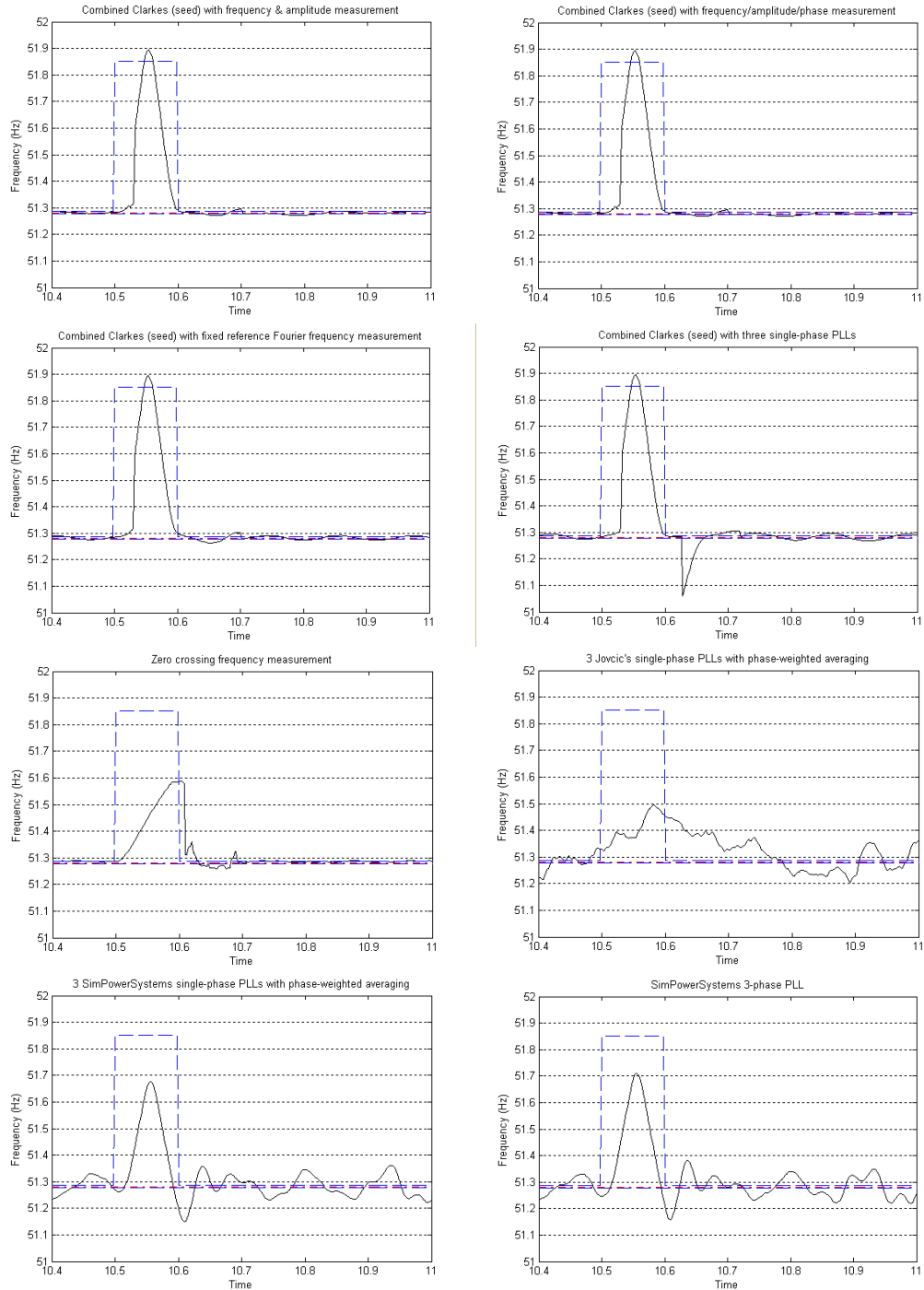
Figure 5-21 : 8 candidate methods; +10 degree phase jump at t=10.5s

Here, the Jovcic PLL shows the least response during the event (mainly because it has a very slow 1.32Hz low-pass filter at its output), but performs badly after the event. The response of the SPS PLLs to the step is limited due to 12Hz/s slew-rate filters which limits their dynamic response. Other algorithms perform adequately.

5.7.5   Results in detail; 40ms 100% three-phase dip to 0pu
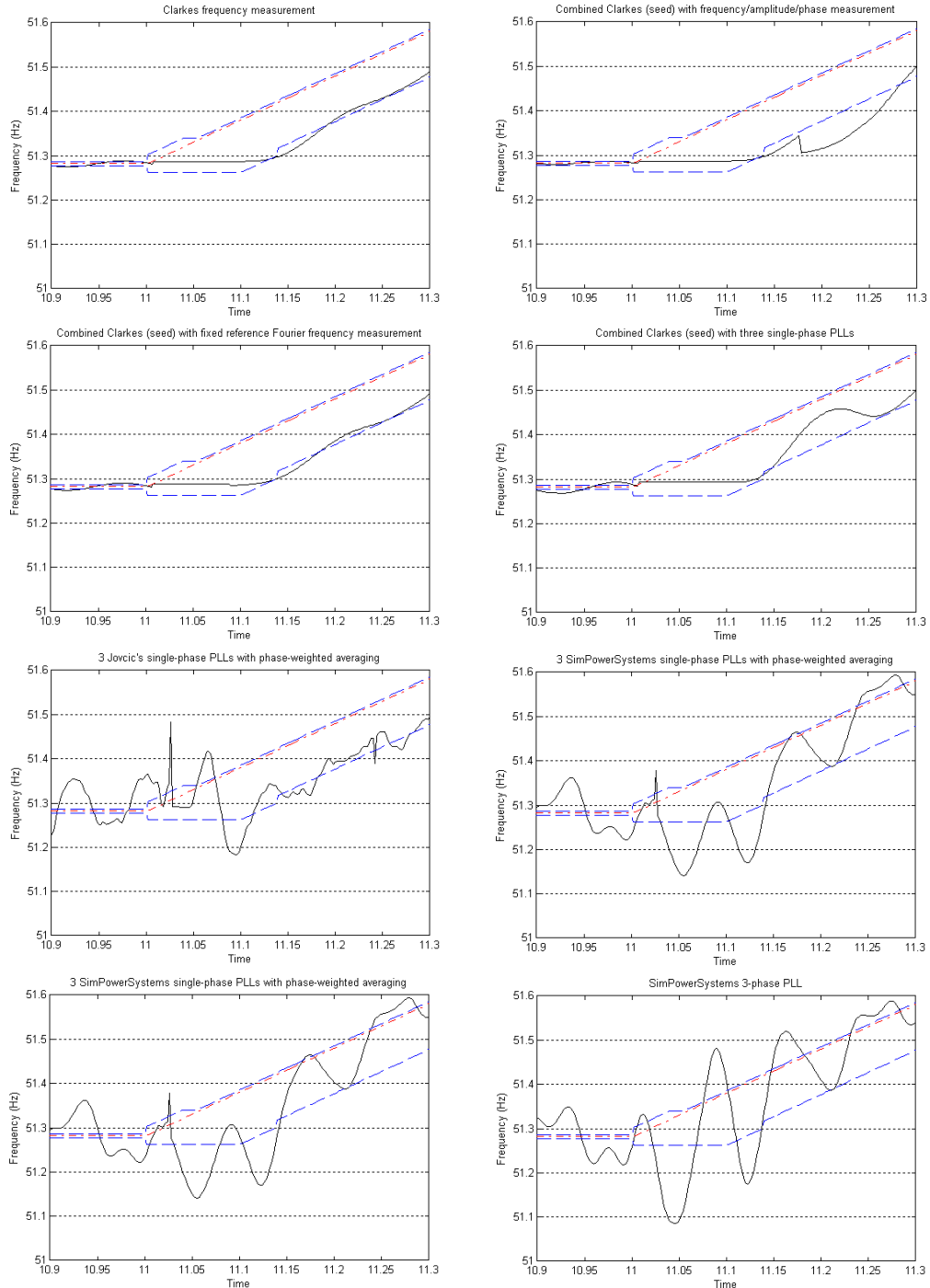
Figure 5-22 : 8 candidate methods; 40ms 100% three-phase dip to 0pu at t=11s

Frequency actually begins to rise during the dip, simulating generator overspeed, but as the dip is 100% deep, there is no way of measuring the signal. The first four algorithms ride through well with a fixed frequency output during the dip. The last four PLL algorithms do not have ride-through and free-run during the fault, which leads to much poorer results.

251

5.7.6 Results in detail; sustained 95% three-phase dip to 0.05pu and 60% three-phase dip to 0.4pu
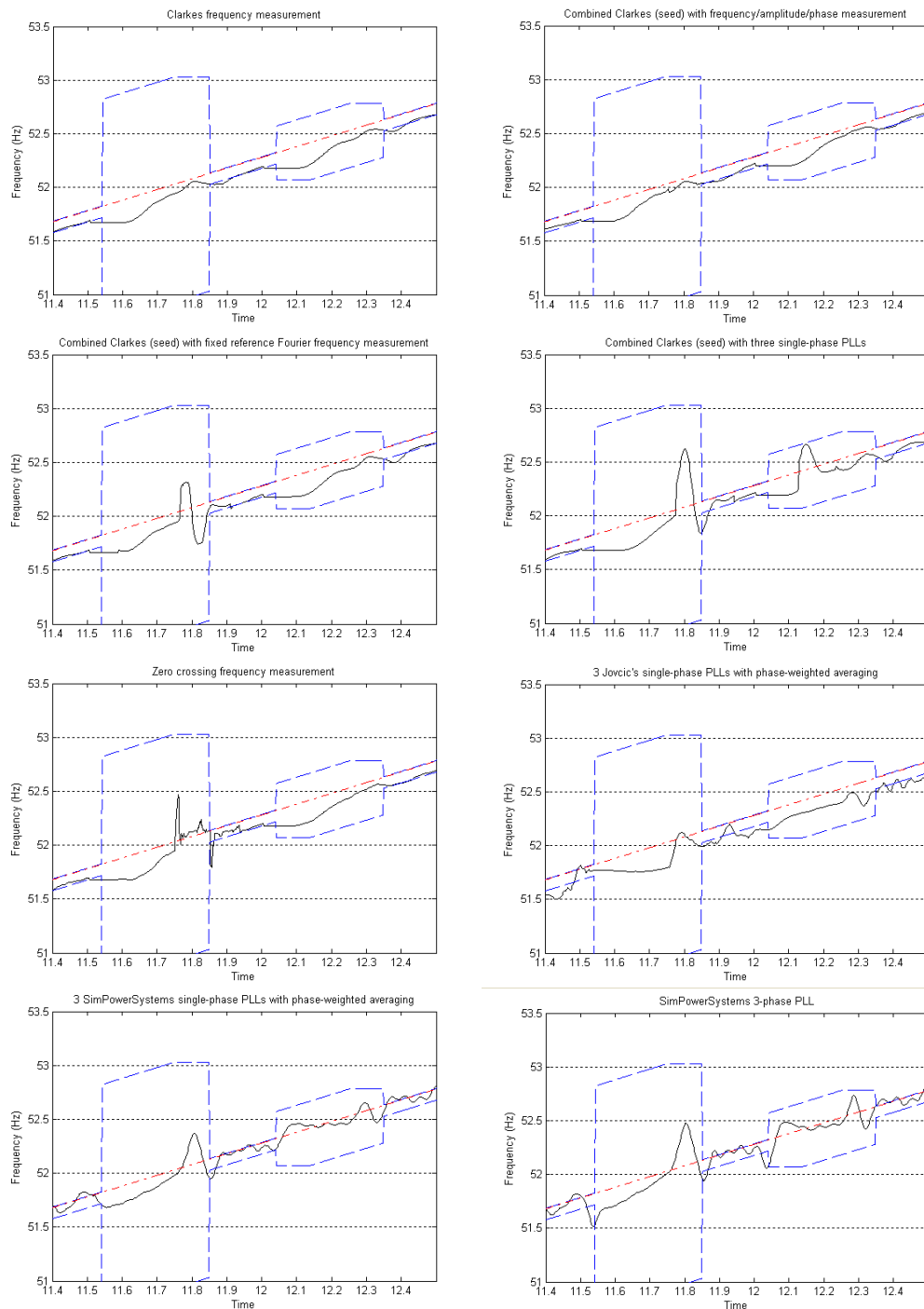
Figure 5-23 : 8 candidate methods; sustained 95% three-phase dip to 0.05pu @ t=11.5s and 60% three-phase dip to 0.4pu @ t=12s

All algorithms perform reasonably during this part of the test waveform, since all three phases are measurable (but the relative noise level is higher). The Clarke transform and the Clarke-FLL hybrid perform the best. Ride-through is briefly activated at the inceptions of the dips.

5.7.7   Results in detail; sustained single-phase fault and two-phase fault
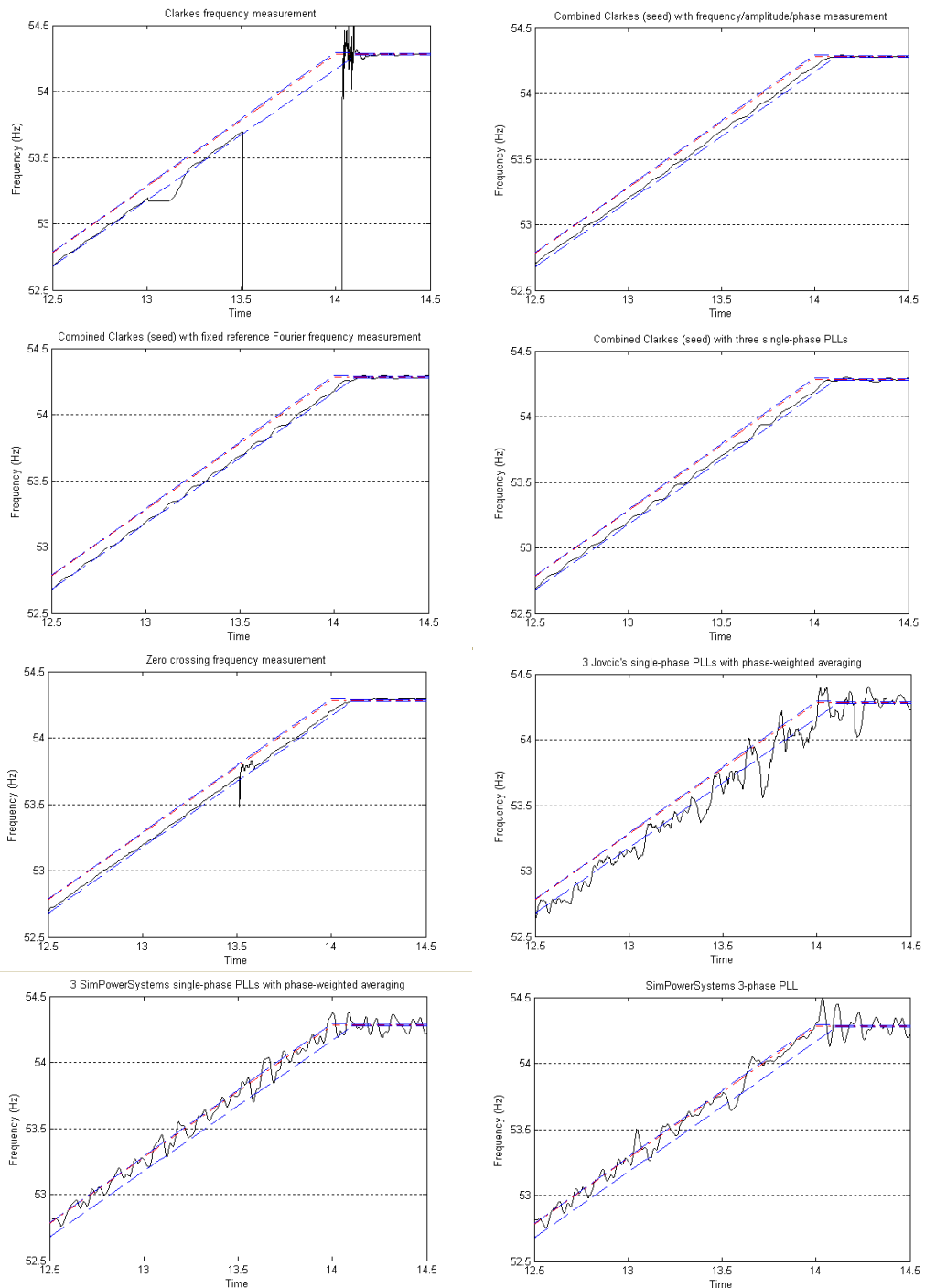
Figure 5-24 : 8 candidate methods; sustained single-phase fault (t=13-13.5s) and two-phase fault (t=13.5-14s)

The Clarke transform measurement becomes invalid during the two-phase fault and gives very poor results. The Clarke-FLL hybrid and the other 2 algorithms developed during this thesis perform well. The zero-crossing, Jovcic PLL, and SimPowerSystems PLLs all show poor performance.

5.7.8   Results in detail; steady state influence qualities at 54.282 Hz
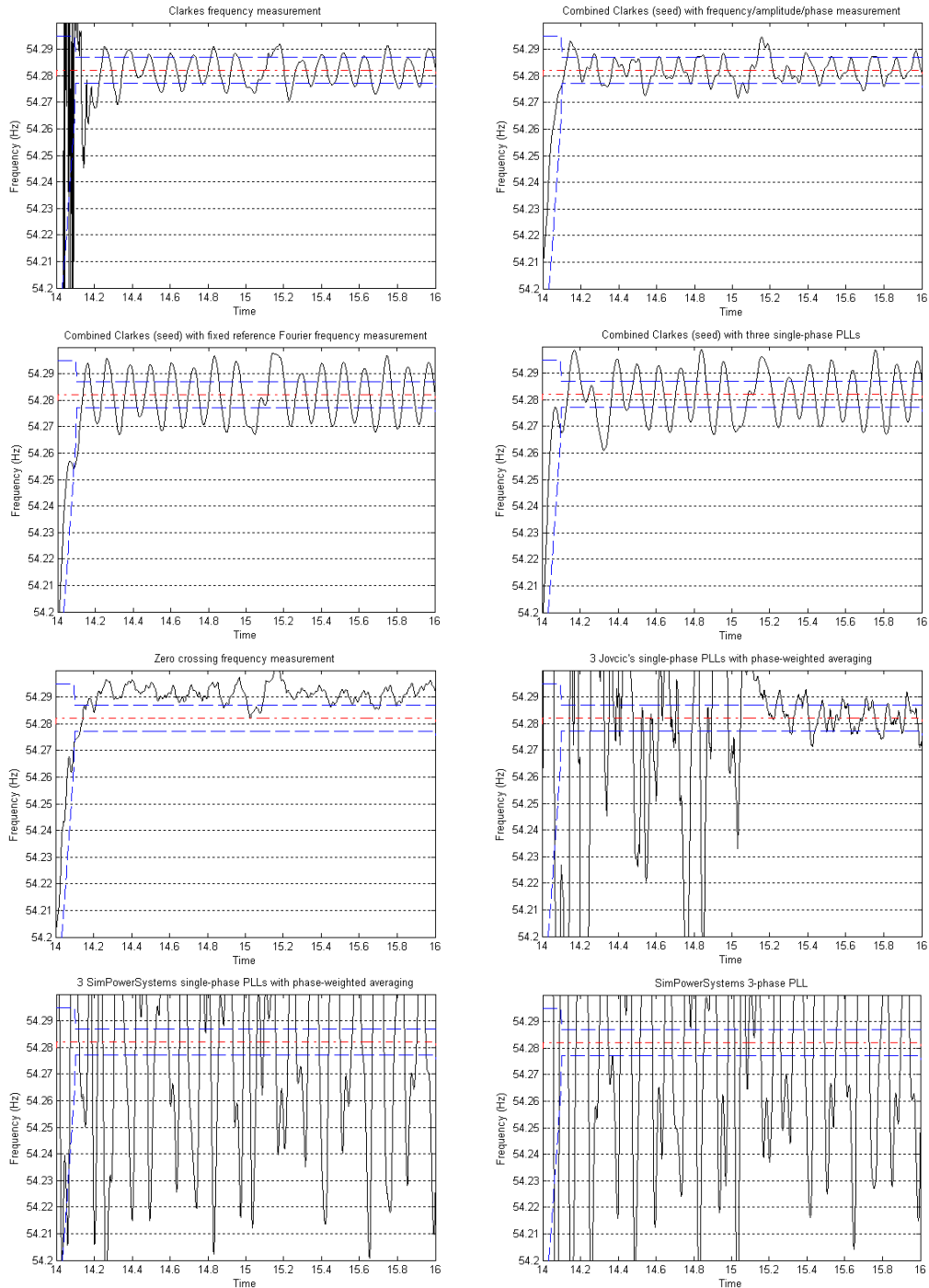
Figure 5-25 : 8 candidate methods; steady state with influence qualities at 54.282 Hz
The results of this scenario are worse than those of Figure 5-20, due to there being only 9.2 samples per cycle, and due to the effect of aliased harmonics. Clearly the Clarke transform and Clarke-FLL hybrid algorithms perform the best. The bias on the zero-crossing measurement is due to inter-harmonics and flicker.

5.7.9 Results in detail; frequency step
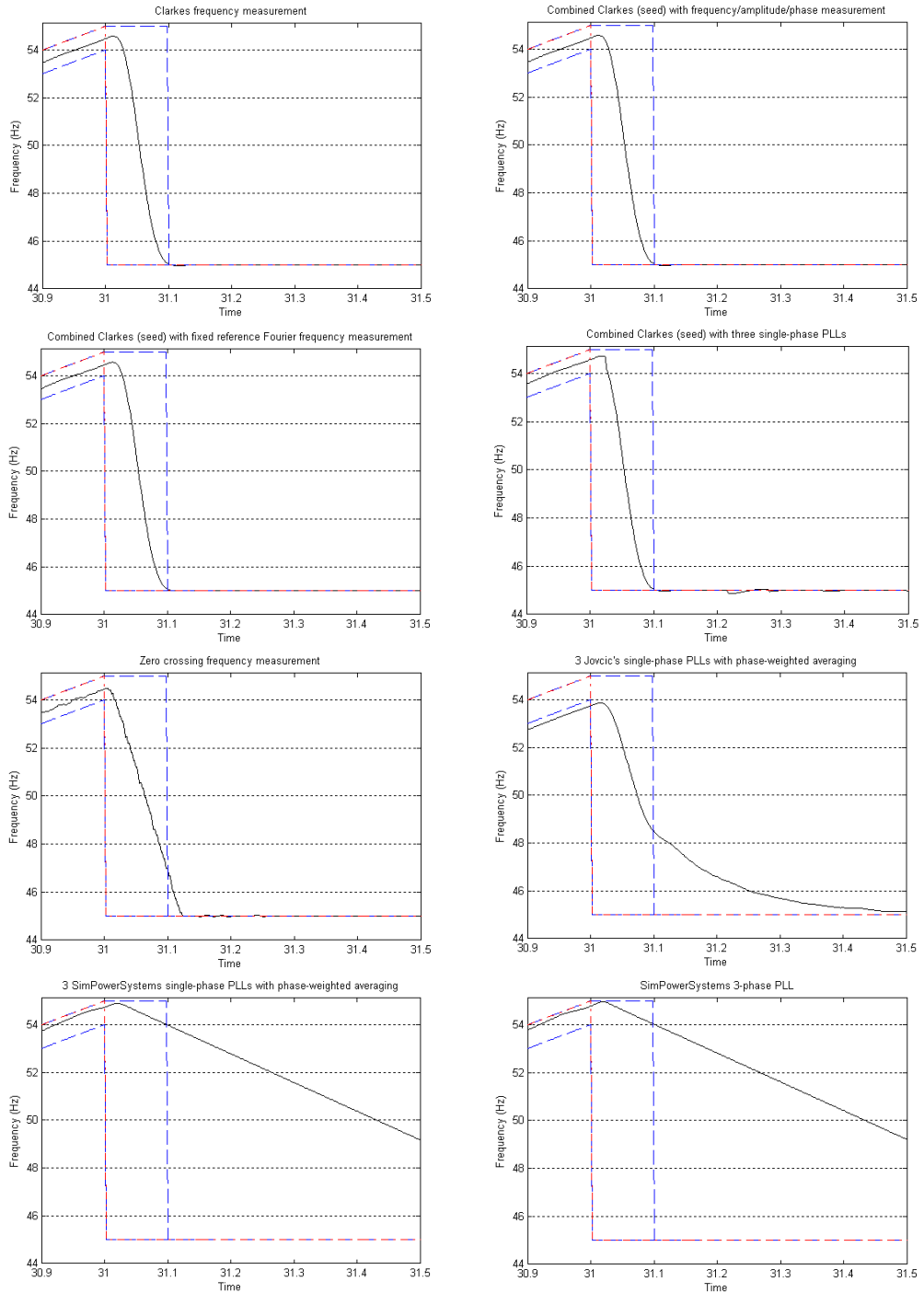
Figure 5-26 : 8 candidate methods; sudden frequency step

This scenario tests the algorithms' response due to disconnection from one network and reconnection to another within a short timeframe. The requirement is for a 5-cycle (100ms) settling time. Clearly the Jovcic's PLL and the SimPowerSystems PLLs apply filtering which does not allow such a step to be followed adequately.

5.7.10 Results in detail; sudden jump from low to high frequencies
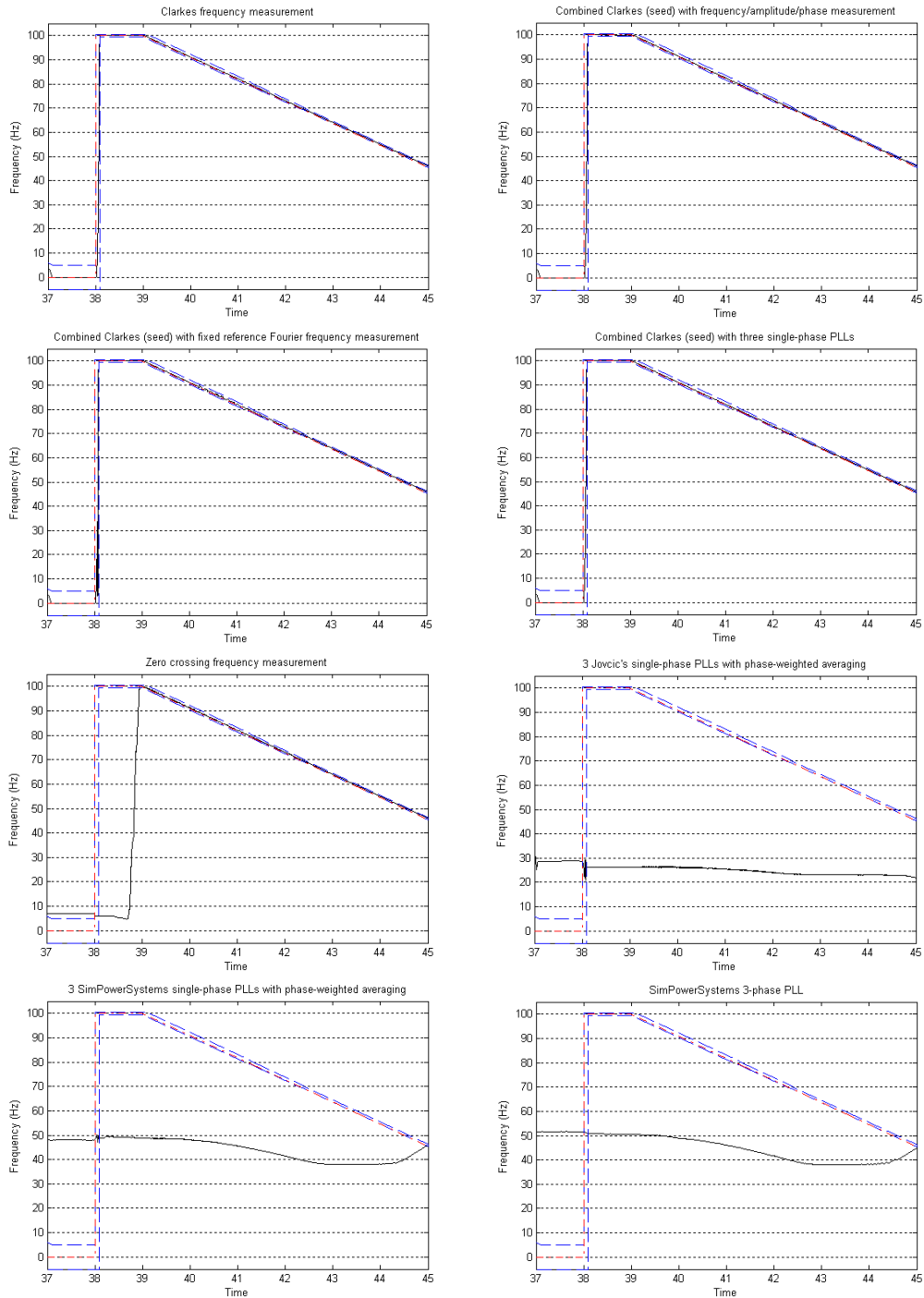
Figure 5-27 : 8 candidate methods; sudden jump from low to high frequencies

The first four algorithms (including the Clarke-FLL hybrid) all seed heavily from the Clarke transform during this scenario, enabling them to track the signal accurately. The Jovcic's PLL and SimPowerSystems PLLs lose lock (before t=37s due to low frequency) and, lacking a seeding algorithm, cannot re-attain it until frequency returns to near-nominal again. In this scenario the Jovcic's PLL does not manage to lock at all during the time period shown.

5.7.11   Results in detail; low frequencies
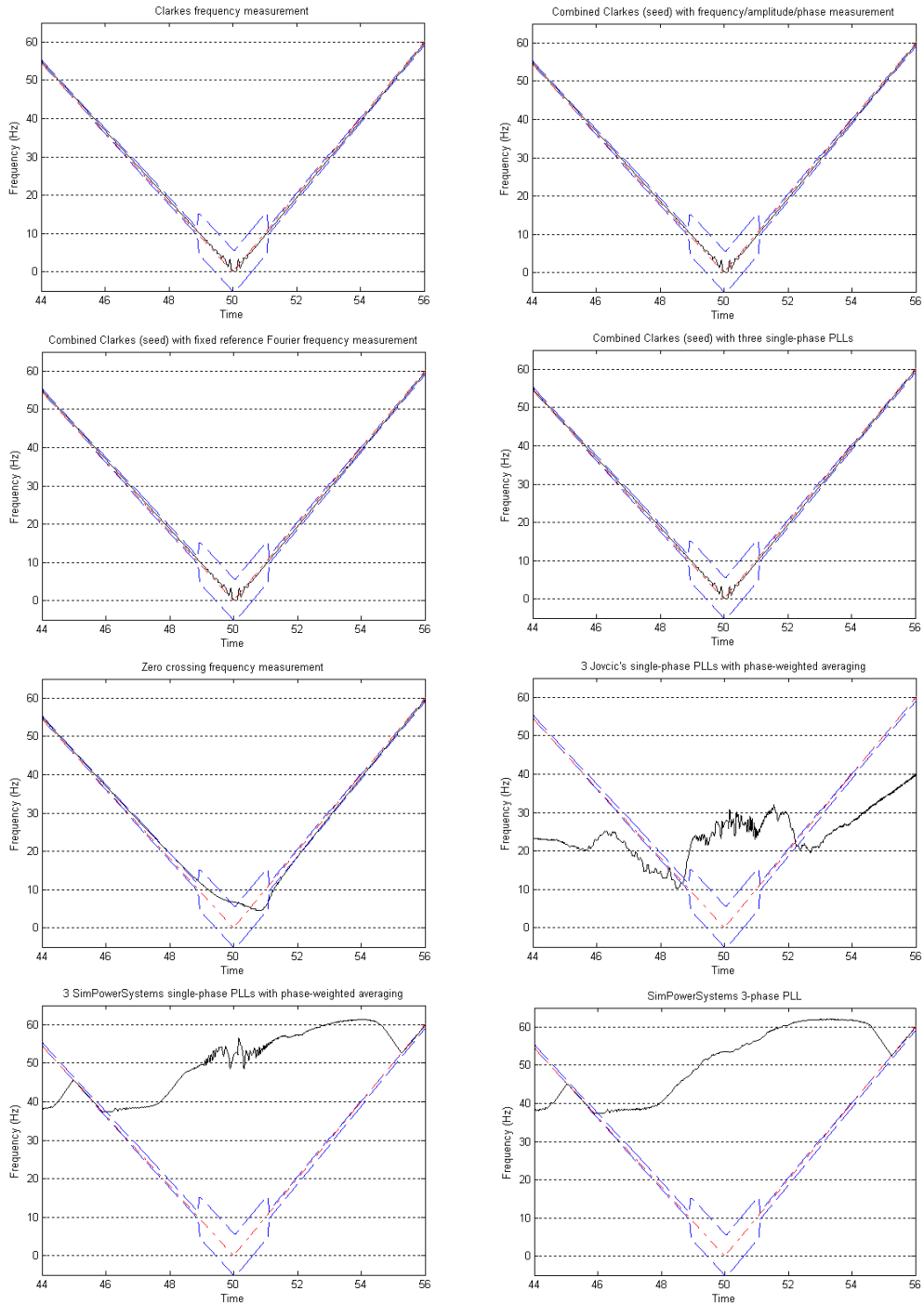
Figure 5-28 : 8 candidate methods; low frequencies

The first four algorithms (including the Clarke-FLL hybrid) all seed heavily from the Clarke transform when frequency drops to levels much less than $f_{nom}/2$, enabling them to track the signal accurately. The Jovcic's PLL and SimPowerSystems PLLs lose lock and, lacking a seeding algorithm, cannot re-attain it until frequency returns to near-nominal again.

5.8   Summary of findings with respect to frequency measurement

A set of specifications for frequency measurement within a microgrid context was laid down in chapter 2. During the course of the work in this thesis, many attempts at optimum frequency measurement algorithms have been made. Not all are presented in this thesis as some proved to be unsatisfactory. The most promising 4 candidates created by the author during the course of this thesis summarised as:-

- An algorithm based upon the Clarke transform, optimised to provide an adaptive output which switches between fast, noisy outputs and slower, more filtered outputs during the 100-200ms time period after a valid set of voltage signals appear. This provides an extremely fast responding, robust measurement which can cope with very low and very high frequencies. It's only significant weakness is its vulnerability to 2-phase faults.

- The Clarke-FLL hybrid, with a set of 3 amplitude/phase measurement blocks at the core. This creates an integrated frequency/amplitude/phase measurement system. The algorithm uses seeding from the Clarke transform measurement when necessary.

- A Fourier analysis method based upon 3 single-phase measurement blocks using a fixed frequency reference at the nominal frequency, plus seeding from the Clarke transform measurement

- A set of 3 single-phase PLLs, together with a seeding mechanism from the Clarke transform measurement

These algorithms all incorporate mechanisms for fault ride-through.

The algorithms have been benchmarked against each other, and against 4 other algorithms for comparison:-

- A zero crossing algorithm
- A set of 3 PLLs to the design of Jovcic (2003)
- The SimPowerSystems 3-phase PLL
- A set of 3 SimPowerSystems 1-phase PLLs

Other algorithms from research literature were considered but did not provide a suitable fit at the low sample rate (500 Sa/s) which was targeted.

The best performing algorithm is the Clarke-FLL hybrid. This algorithm achieves by far the lowest error score in the performance tests (see Table 5-2), and this also shows up in the traces on Figure 5-19 to Figure 5-28. This algorithm has a much better locking characteristic than a PLL when only frequency (and not phase) needs to be measured.

The optimal architecture for the Clarke-FLL system re-uses many of the concepts originally explored and implemented in chapters 3 & 4. This not only leads to an algorithm with very good performance, but also allows substantial code re-use.

For the worst-case microgrid voltage waveforms with 53% THD_V, plus unbalance, inter-harmonics, and flicker,(see section 2.7), it has been found possible to achieve the tightest target specification of ±0.005 Hz in all but a few transient events and at the worst frequencies, with an allowed response time of 100ms (nominally 5 cycles). Even during the transient events and at the worst frequencies, the ±0.005 Hz specification is only narrowly missed, and a ±0.01 Hz peak error specification is almost always achieved. This provides a significantly more robust and faster measurement than specified by BS EN 61000-4-30 (BSI, 2003) for a "Class A" instrument, with twice the accuracy, as shown on Table 5-3. This is despite the low frame rate used within the major algorithms.

	BS EN 61000-4-30 "Class A" performance (BSI, 2003)	Performance of architecture developed in this thesis, worst case scenario
Scenario	Harmonics up to 20% THD (twice times BS EN 61000-4-3 table 5, class 3) (BSI, 2002)	THD up to 53%. 2 cascaded low-pass anti-aliasing filters with cut-off frequency set to 125 Hz. Instrumentation noise level (before the ADC) at or below 0.005pu RMS (46dB SNR). A 12-bit ADC with a bit noise of no more than 2 bits RMS.
Frequency measurement	±0.01 Hz (±0.02%, ±0.0002pu) A measurement time of 500 cycles (10s)	±0.005 Hz (±0.01%, ±0.0001pu) A settling time of 5 cycles (100ms) during transients and a total averaging time of 9 cycles (180ms) normally

Table 5-3 : Performance of frequency measurement architecture proposed by this thesis, versus standard "Class A" performance

Applying this new algorithm to microgrid control systems and generator management systems may lead to the following benefits, relative to other frequency measurement methods:-

- Quicker settling time, hence more reliable information for decision-making algorithms and relays etc.

- Lower measurement latency. Hence, less likelihood of power system oscillations building up due to unintentionally lagged controls.

- Functionality at low sample rates, allowing the algorithm to be deployed on cheap microcontroller platforms in conjunction with other measurement/control algorithms.

- Lower measurement ripple in the presence of influence qualities such as harmonics, inter-harmonics, noise, unbalance and flicker. This leads to smaller power system oscillations due to less ripple being fed through to droop controllers etc...

- The new frequency measurement will provide a much better basis for a loss-of-mains relay based upon ROCOF, or an under/over-frequency relay, due to the combination of the above improvements. This is explored further in chapter 6.

5.9   References for chapter 5

Aghazadeh, R. et al. (2005). New technique for frequency and amplitude estimation of power system signals. IEE Proceedings on generation, transmission and distribution, vol. 152, no. 3, pp 435-440.

Awad, H., Svensson, J. & Bollen, M. (2005). Tuning software phase-locked loop for series-connected converters. IEEE transactions on power delivery, vol.20 issue 1. pp 300-8.

BSI, British Standards. (2000). Voltage characteristics of electricity supplies by public distribution systems. BS EN 50160:2000. ISBN 0 580 35925 5

BSI, British Standards. (2002). Electromagnetic compatibility (EMC) — Part 2-4: Environment — Compatibility levels in industrial plants for low-frequency conducted disturbances. BS EN 61000-2-4:2002. ISBN 0 580 40489 7

BSI, British Standards. (2003). Electromagnetic compatibility (EMC) – Part 4-30: Testing and measurement techniques – Power quality measurement methods. BS EN 61000-4-30. ISBN 0 580 42081 7

Canteli, M. et. al. (2006). Three-phase adaptive frequency measurement based on Clarke's transformation. IEEE transactions on power delivery vol.21 issue 3. pp 1101-5.

Choi, J., Kim, Y. & Kim, H. (2006). Digital PLL control for single-phase photovoltaic system. IEE proceedings on electric power applications, vol.153 issue 1. pp 40-6.

Chung, S. (2000). Phase-locked loop for grid-connected three-phase power conversion systems. IEE proceedings on electric power applications, vol. 147 issue 3. pp 213-19.

Clarke, E. (1943). Circuit analysis of a-c power systems. Wiley.

Dash, P. et. al. (2000). An extended complex Kalman filter for frequency measurement of distorted signals. IEEE transactions on instrumentation and measurement, vol.49 issue 4. pp 746-53.

Han, B. & Bae, B. (2006). Novel phase-locked loop using adaptive linear combiner. IEEE transactions on power delivery Han, vol.21 issue 1. pp 513-14.

Ignatova, V. et. al. (2005). Classification and characterisation of three-phase voltage dips by space vector methodology. International conference on future power systems. P 1600556.

Johns, A. & Salman, S. (1995). Digital protection for power systems. IEE Power Series 15, ISBN 086341303X.

Jovcic, D. (2003). Phase locked loop system for FACTS. IEEE transactions on power systems vol.18 issue 3. pp 1116-24.

Lin, T. & Domijan, A. (2005). On power quality indices and real time measurement. IEEE Transactions on power delivery, vol. 20 no. 4, pp 2552-2562.

Lopez, A. et al. (2008). Power system frequency measurement under nonstationary situations. IEEE Transactions on power delivery, vol. 22 no. 2, pp562-567.

Moore, P., Carranza, R. & Johns, A. (1996a). Model system tests on a new numeric method of power system frequency measurement. IEEE Transactions on power delivery, vol. 11 no. 2, pp 696-701.

Roscoe, A. (2005). Systems and methods for performing analysis of a multi-tone signal. US Patent number US 6,873,923. March 29th, 2005.

Salcic, Z., & Mikhael, R. (2000). A new method for instantaneous power system frequency measurement using reference points detection. Electric power systems research vol.55 issue 2. pp 97-102.

6   Loss-Of-Mains detection

Within this chapter, the frequency measurement algorithm developed in chapter 5 is integrated within a novel new relay for loss-of-mains (LOM) detection. Initially, this relay is investigated in simulation (see section 6.1), and then using real data from captured network events in section 6.2. Finally, in section 6.3, the LOM-detection relay is augmented by a novel microgrid management strategy. This strategy makes small adjustments to the reactive power flow between a microgrid and its parent network. This allows an exact active power match within the microgrid (generation to load), while still avoiding the non-detection zone (NDZ) of the LOM-detection relay. The combined measurement and management algorithms are embedded in a real-time microcontroller platform and tested in a real microgrid power system at the 2kVA-100kVA scale.

Previous work at the University of Strathclyde has analysed the performance of several commercially-available LOM (loss-of-mains) relays and their performance (Dysko, 2006, 2007, 2007b). A LOM event occurs when an upstream fault causes disconnection of a local power system from a parent network. The disconnection can be caused by a permanently or temporarily severed line or cable, or by a short circuit fault which causes a permanent or temporary circuit-breaker trip.

Detection of LOM is important for two main reasons, when generation sources are distributed within a power network:-

- To ensure safety against electric shock, LOM must be detected within a reasonable timeframe, such that generators within islanded networks will be disconnected or managed appropriately in a deliberate islanded state.

- To avoid damage to generators and distribution equipment, loss-of-mains should be detected such that subsequent reconnections to the parent network do not occur when the distributed generator(s) are out of synchronisation with the parent network.

- If LOM is not detected and a generator continues to operate in grid-connected mode, excursions to frequency and voltage may result which might violate power quality expectations and cause damage to loads.

During normal operation, distributed generators are conventionally operating in grid-connected mode, with real and reactive (P & Q) power output targets set at the generator, in combination with appropriate droop controls. When a loss-of-mains event occurs, the generators are no longer connected to the parent network. In this

configuration, the grid-connected control strategies will not generally keep the frequency and voltage of the power island at the same pre-disconnection values. The overall system may stabilise at some new (and acceptable) values of frequency and voltage, or the frequency and/or voltage may rise or fall outside acceptable limits, resulting in disconnection. The worst case generally presented for LOM detection is where the local real and reactive loads are, by chance, exactly (of very almost) equal to the local generation target outputs. In this case, in theory, the system could continue to operate at the same frequency and voltage, in which case the system is said to be within the "non detection zone" (NDZ) of the LOM relay. In most practical cases, however, there will be some load/generation imbalance in either P or Q, and any imbalance will cause both frequency and voltage excursions. The bigger the imbalance, the bigger the excursions.

The report by Dysko (2006) analyses different types of conventional LOM-detection relays under different load/generation imbalances and different non-LOM fault conditions, to assess their sensitivity (when local generation is closely matched to local loads) and discrimination (against non-LOM fault conditions, noise, harmonics, flicker, transients etc.). In this case, the non-LOM faults considered are limited to single, two and three-phase faults at various positions with a distribution network. The discrimination against non-LOM disturbances causing spurious tripping of LOM relays has been identified as a major current problem with many commercial and proposed LOM detection relays (Dysko, 2007).

In this section of the thesis, a new algorithm for detecting the LOM condition, based purely on local passive measurements of voltage, is proposed. Later, in section 6.2, a novel DG control strategy is also proposed and tested. The combination of the relay and the control strategy allows detection of LOM within 1-2 seconds during all scenarios, by deliberately just avoiding the non-detection-zone (NDZ) (Ye, 2004) of active and reactive power balance.

Generally the commercially available passive LOM-detection relays fall into two types: ROCOF (Rate of change of Frequency) and Vector Shift. Vector-shift relays are analysed in Dysko (2006) & Freitas (2005), and are found to be less suitable due to their relative inability to detect genuine LOM events. The ROCOF relays perform better, but when choosing the trip settings for commercial ROCOF relays, dilemmas are faced when trying to meet the demands of both sensitivity and discrimination. This is the reason that ROCOF relay settings in practice are set as low as 0.1 Hz/s, but up to 1.2 Hz/s (Vieira, 2006c), while the natural frequency variations of the grid to which these relays are connected are

much less than 1.2 Hz/s (see Table 2-1 and section 6.2). A review of other candidate passive LOM detection strategies is given in Dysko (2007), which highlights the fact that most passive LOM detection strategies can be made to be very sensitive, but few authors have paid attention to the issue of discrimination and spurious tripping. The new methods proposed in Dysko (2006) provide better results than commercially available ROCOF and vector-shift relays, but still a single setting has not be found for these algorithms which achieves both the sensitivity and discrimination required. (Vieira, 2006a) examined the use of under-frequency and over-frequency relays to detect loss-of-mains. Such a relay is good at avoiding spurious trips, but has the disadvantage of having to wait until the frequency excursion is quite large before detecting the loss-of-mains condition. This means that ride-through action (trying to switch to a stable and deliberate power island) after detection will be hard, as the frequency will be verging on allowed limits at the instant of changeover.

It should also be mentioned that there are many works concerning active loss-of-mains methods which are specific to inverter-connected generation. These techniques involve injection of current waveforms with deliberate harmonic content, either in pulsed or steady-state modes. Good examples of these are Huang (2001), Timbus (2004), and Sumner (2004a, 2004b). These methods require specific inverter hardware or current injection apparatus and are "active" forms of detection. Such methods are dependent upon the injection hardware functioning correctly and can therefore be regarded as less reliable than "passive" methods such as that proposed in this work. Where many such active devices are placed in the network, further risks arise such as degradation of power quality, and interference between the multiple current injection waveforms. The work in this thesis is targeted at strategies for "passive" detection of loss-of-mains which are independent upon generator type, and so these methods are not applicable.

6.1   Application to loss-of-mains detection (simulations)

6.1.1   Proposed new loss-of-mains detection algorithm – Phase Offset Relay (POR)

This thesis proposes a new algorithm, based most closely upon the principle of "Method 2" given by Dysko (2006). There are also some parallels (and many differences) to a method proposed by Wall (2004). Compared to the method proposed by Dysko (2006), the proposed algorithm contains the following major differences:-

- the robust integration algorithms of section 3.2 are used, avoiding integrator wind-up

- a triggering/resetting subsystem to avoid constant tripping, in place of the

high-pass filter used by Dysko(2006). This allows setting of the trigger threshold by a meaningful parameter, the expected ROCOF levels during normal system operation, instead of by a high-pass filter cutoff frequency which has little obvious physical significance.

- An allowance for variable phase offset trip thresholds during and immediately following faults. This significantly enhances the discrimination of the relay against faults and disturbances, without reverting to a (dangerous) complete blocking of the trip signal.

The new algorithm is called a Phase Offset Relay, or POR.

The relay operates on the principle that the phase offset of a system relative to a stable frequency can be estimated by

$$\phi = 2\pi \int \left(\int ROCOF \cdot dt \right) \cdot dt = \pi \cdot ROCOF \cdot t^2$$ where Φ is in radians

(6.1)

or

$$\phi = 360 \int \left(\int ROCOF \cdot dt \right) \cdot dt = 180 \cdot ROCOF \cdot t^2$$ where Φ is in degrees

(6.2)

Equation (6.2) can be inverted to reveal trip times for given (constant) ROCOF values and trip thresholds:

$$t = \sqrt{\frac{\phi}{180 \cdot ROCOF}}$$ where Φ is in degrees

(6.3)

The advantages of this algorithm over a ROCOF algorithm are:-

1. Since the measurement of frequency is "hard" to make without noise on the measurement, the differentiation of frequency to obtain a value of ROCOF leads to an even noisier measurement. Using such a noise measure directly within a relay increases the risk of spurious trips (or non-detection of events). By taking the double-integral (equivalent to a double-averaging) of ROCOF, the noise is substantially reduced.

2. The double-integral stage also gives an answer with physical significance, i.e. the approximate phase deviation of the local voltage measurement from the steady-state value. This phase represents the risk of damage due to out-of-phase re-synchronisation.

3. The trip time for traditional ROCOF relays is generally a fixed time; i.e. >0.2Hz/s for 200ms must be measured for a trip. This neither allows tripping in less than 200ms when ROCOF is large, nor tripping in more than 200ms when ROCOF may be only slightly higher than 0.2Hz/s. It should be noted that a typical ROCOF relay setting of 0.2Hz/s for 0.2 seconds corresponds to a phase offset of only 1.4° which would be of no concern should an auto-reclose action occur. Given that up to 2 seconds is allowed under IEEE 1547 (IEEE, 2003)[1] for LOM detection, at a rate of 0.2Hz/s a trip time of 745ms would be more appropriate since this would be the time at which a 20° phase offset occurred between local generation and the parent network (by equation (6.3)). The phase offset relay does exactly this, using a fixed angle setting but a flexible trip time. This different approach allows much better discrimination between genuine LOM events and other disturbances such as normal load steps which cause brief frequency deviations.

The inner core of the algorithm is shown here mathematically. The real implementation is all coded digitally in discrete time steps.

1. Measure frequency f (by the algorithms of chapter 5)

2. Calculate $ROCOF = \dfrac{df}{dt}$. This can be done on a two-sample basis. Noise and/or spikes on this signal is acceptable as it will be averaged out by two cascaded integration steps.

3. Calculate frequency offset $\Delta f = \int_{t_t}^{t} ROCOF \cdot dt$ where t_t is the time at which the relay triggering starts. Note that Δf is equivalent to the measured value f, with the DC component removed. The differentiation and integration stages are a convenient way of accomplishing this, while also producing a value of ROCOF for indication (although it may be noisy and takes no direct part in the relay operation).

---

[1] The UK equivalent document, ER G59/1 (ENA, 1991), says only that "The setting of the relays should be agreed with the PES". ETR 113 (ENA, 1995) states than detection within 1 second may be required in some UK rural distribution networks due to potential auto-reclose times in feeders, but this is not a general requirement across the UK. ETR 113 also gives no indication of practical relay settings for DG installations with potential DG-to-load power matches of less than 0.1pu which enable such detection to be practically achieved using a ROCOF or similar relay.

4. Calculate phase offset $\phi = 360 \int_{t_t}^{t} \Delta f \cdot dt$ where t_t is the time at which the relay triggering starts, and Φ is the phase offset in degrees.

5. Check the magnitude of the phase offset, abs(Φ) against the trip threshold Φ_{Trip}. If the trip threshold is exceeded, a trip signal is generated. No qualifying time is required since the double-integrated signal is very clean and free from noise, and the initial triggering algorithms (see below) remove spurious trips due to normal fluctuations of the network frequency due to load changes, generator despatching, switching etc..

A triggering algorithm is also required; otherwise the integrals above would lead to tripping due to the normal small fluctuations in frequency which occur in all power networks. Adding the triggering and implementing the entire algorithm requires a number of parameters and variables to be set and calculated. These are tabulated below for clarity.

Symbol	Meaning	Typical parameter value (or calculated variable)	Derivation
f	Frequency	Measured	typically 40-70Hz
ROCOF	Rate of Change of Frequency	Calculated	df/dt (two-sample basis)
Δf	Frequency variation	Calculated	$\Delta f = \int_{t_t}^{t} ROCOF \cdot dt$
t	Time now	Implicit	
t_t	Time when the relay is triggered	Implicit	
Φ	Phase offset (degrees)	Calculated	$\phi = 360 \int_{t_t}^{t} \Delta f \cdot dt$
Φ_{Trip}	Phase offset trip setting	Typically 20 degrees	Set such that an accidental resynchronisation does not cause excessive currents. For a generator with leakage reactance 0.1pu, a 20 degree synchronisation results in currents of approximately sin(20°)/0.1=3.4 pu. Also, this setting determines trip times.
f_{Nom}	Nominal frequency	Typically 50Hz or 60Hz	

$R_{Trigger}$	Trigger ROCOF threshold	Typically 0.2 Hz/s	Set by the expected ROCOF during normal network operation (not accounting for close-in faults but allowing for distant faults)
$N_{TriggerCycles}$	Number of nominal cycles over which to evaluate triggering window	Typically 5	Set as high as possible, with the limit that the shortest LOM trip time will then be $t_{Trigger}$
$t_{Trigger}$	Trigger window time	Calculated (constant)	$N_{TriggerCycles}/f_{Nom}$
$\theta_{Trigger}$	Triggering phase angle threshold (degrees)	Calculated (constant)	$360.R_{Trigger} \cdot t_{Trigger}^2$
Δf_T	Frequency change during trigger window	Calculated	Rolling definite integral of ROCOF over last $t_{Trigger}$ seconds
θ	Phase change during trigger window	Calculated	360 times rolling definite integral of Δf_T over last $t_{Trigger}$ seconds

Table 6-1 : Parameters and variables for the Phase Offset Relay (POR)

The triggering threshold is set by an estimate of the expected upper level of expected ROCOF, $R_{Trigger}$, and a setting parameter $N_{TriggerCycles}$ that defines a time period $t_{Trigger}=N_{TriggerCycles}/f_{Nom}$ over which the triggering signal is evaluated. Over this timeframe, the steps 3-4 (above) are evaluated using rolling time windows of fixed time length $t_{Trigger}$, to calculate the frequency offset Δf_T and phase offset θ which has occurred over the last $t_{Trigger}$ seconds. This can then be checked against a phase offset trigger threshold $\theta_{Trigger}$ which would occur due to a constant ROCOF of $R_{Trigger}$.

At first glance, $\theta_{Trigger}$ would be derived directly from (6.2), given $t_{Trigger}$ and $R_{Trigger}$. but here the value of $\theta_{Trigger}$ is determined in a slightly different manner. This is because the triggering is evaluated over rolling windows of constant length $t_{Trigger}$ seconds, where $R_{Trigger}$ is also constant. Thus, evaluating

$$\theta_{trigger} = 360\int_0^{t_{Trigger}}\left(\int_0^{t_{Trigger}} R_{Trigger} \cdot dt\right).dt = 360 \cdot R_{Trigger} \cdot t_{Trigger}^2$$

(6.4)

where $\theta_{Trigger}$ is in degrees, i.e. double the value suggested by (6.2)

As well as triggering the main integrations within the trip detection algorithms, the triggering subsystem also has the power to reset and zero the trip detection algorithms. This means that sometimes the relay may be triggered (the main integrators of steps 3 and 4 above may begin to accumulate frequency and phase offset amounts), but will not trip. These events will occur when measured ROCOF rises above the trigger level $R_{Trigger}$ but the

subsequent frequency disturbance is not sufficient to cause an actual trip by violation of $\Phi > \Phi_{Trip}$ or $\Phi < -\Phi_{Trip}$. Such functionality is crucial to the discrimination function of the relay, to avoid spurious trips.

In the algorithm presented here, the triggering subsystem causes such a reset when the value θ evaluated over the rolling window of the triggering subsystem undergoes a zero crossing. Because the values of θ are evaluated using double integration of the ROCOF signal, these zero crossings are well defined and relatively noise-free. An alternative method considered was to reset the subsystem when the ROCOF value passed through a zero crossing, which in some ways is more desirable. (When ROCOF passes through 0, the frequency must be either flat, or at a maxima/minima, implying that frequency is under control and not crashing upwards or downwards). However, since the ROCOF measurement can be noisy, this method was not chosen. The other alternative considered was to detect zero-crossings of the Δf_T signal. In practice the times at which the resets would occur due to any of the 3 methods is found to be about the same, so the most noise-immune method is chosen.

The algorithm details are shown below in Figure 6-1. In addition, some external code (Figure 6-2) can be used to widen the phase offset trip threshold Φ_{Trip}, during and immediately subsequent to serious (close-in) balanced and unbalanced faults which do not lead to LOM events. This code widens the trip setting from the normal setting to a much wider setting during serious balanced and unbalanced faults. When the fault condition is lifted, the trip threshold slides back at a constant rate to the normal setting, over a period of time. This allows for significant power system oscillations within the immediate post-fault period, and provides a solution to the conflicting demands of sensitivity and discrimination which are identified (but not solved) in Dysko (2006 & 2007).

It should be noted that some LOM relays block the trip signal entirely (Dysko (2007), Freitas (2005) & Vieira (2006c)) when such fault conditions are detected, whereas the solution proposed here is only a widening of the angle trip threshold. Thus, with the algorithm below, the LOM relay will still throw a trip if the local frequency diverges consistent with a genuine LOM event, potentially triggered by a fault condition. The aim is to allow a certain amount of rotor swinging and power system oscillation (which can exceed the standard trip setting of 20°) after a fault without throwing a LOM trip, but still to have a wider trip setting in place which will be exceeded if (for example) pole slip or an actual LOM event occurs.

Phase offset relay Andrew Roscoe, 2006-2008

Integrates the ROCOF to give a frequency offset from the (assumed stable) grid frequency.
Further integrates the frequency to obtain the phase offset of the measured system from the (assumed stable) grid.
Generates a trip theshold when the maximum tolerable phase offset (about 20 degrees) is exceeded.

0
double
[Zero]
1
ROCOF
[ROCOF]

[ROCOF]
Signal   Running average
TickTock, Fixed Tint, no interp
TickTock, Fixed Tint, no interp   TickTock, Fixed Tint, no interp   Integral over Tint
Tint = TriggerCycles/Fnom   Avg/int, fixed Tint, no interpolation, opt
Tint = TriggerCycles/Fnom
0
double
[Zero]
[FrequencyOffsetNCycle]

[FrequencyOffsetNCycle]   360   Signal   Running average
TickTock, Fixed Tint, no interp   Integral over Tint
Avg/int, fixed Tint, no interpolation, opt1
Tint = TriggerCycles/Fnom
0
double
[Zero]
[PhaseOffsetNCycle]

2
ZeroROCOFAndIntegrators   [Zero]

4
Phase offset to trip (deg)   [Trip_Phase]

PhaseOffsetNCycleTriggerLevel=ROCOF_min*360*(TriggerCycles/Fnom)^2;
% Note, this is NOT ROCOF_min*360*(TriggerCycles/Fnom)^2/2, ie
% the /2 is dropped. This is because the FrequencyOffset5Cycle
% parameter is evaluted over a rolling t=5/Fnom window and
% as such is constant for a constant ROCOF input, whereas the
% theoretical integration normally has the frequency gradually
% increasing as the PhaseOffset integrates.

360*(TriggerCycles/Fnom)^2
double
5
Minimum ROCOF to cause trip   ×   [PhaseOffsetNCycleTriggerLevel]

[PhaseOffsetNCycle]   x   abs(x)
abs
[PhaseOffsetNCycleTriggerLevel]   >   Switch on   Latched output
Switch off
Latched boolean (priority off)2   [Integrate]
Boolean Pulse   [StartToIntegrate]
0-1 detect
Signal   Zero crossing
Direction
Zero crossing detector

[StartToIntegrate]
[Integrate]   NOT   OR   [ResetIntegrators]

[ROCOF]
[ResetIntegrators]   y
Reset (when 1)Int y dt   [FrequencyOffset]
[FrequencyOffsetNCycle]   reset value
[StartToIntegrate]   Trapezial integrator
with reset and limits
0   -Fnom <= I <= Fnom
double

[FrequencyOffset]   360
[ResetIntegrators]   y
Reset (when 1)Int y dt   [PhaseOffset]
[PhaseOffsetNCycle]   reset value
[StartToIntegrate]   Trapezial integrator
with reset and limits1
0   -180 <= I <= 180
double

[PhaseOffset]   x   abs(x)
abs1   >   Switch on   Latched output   1
[Trip_Phase]   Switch off   Trip
3   Latched boolean (priority off)
ResetTrip

[FrequencyOffsetNCycle]   FrequencyOffsetNCycle
[PhaseOffsetNCycle]   PhaseOffsetNCycle
[PhaseOffsetNCycleTriggerLevel]   PhaseOffsetNCycleTriggerLevel
[Integrate]   Integrate   2
[FrequencyOffset]   FrequencyOffset   Extra outputs
[PhaseOffset]   PhaseOffset

Figure 6-1 : Phase offset Relay (POR); algorithm detail

Figure 6-2 : POR; adaption of trip threshold immediately after faults

An example of the application of this algorithm is shown below in Figure 6-3, which is reproduced here from the simulation results of section 6.1.3.3 presented later. The blue line shows the apparent nodal phase variation due to power system oscillation immediately after fault clearance. The dashed red line shows how the trip setting can be automatically widened when the fault is detected (to 100° in this case), and then tapers back to the nominal setting (20° in this case) over a number of seconds after the fault is cleared.
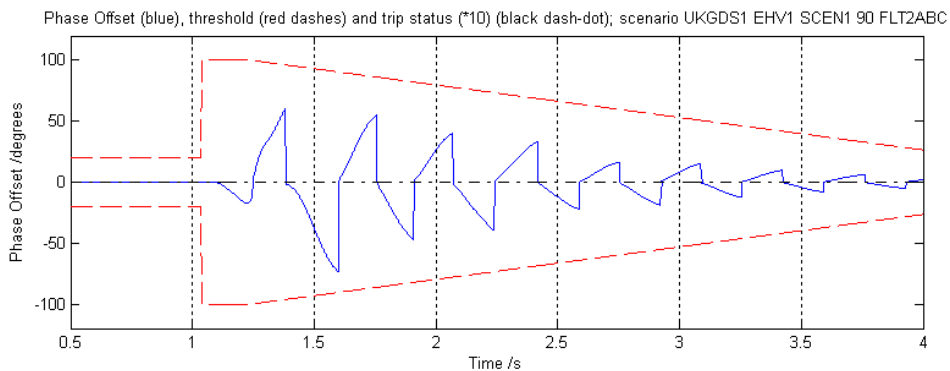
Figure 6-3 : POR; adaption of trip threshold immediately after faults

An additional algorithm to explicitly temporarily disable the trip setting could also be included in the future, although it is not included in this thesis. Veira (2006b), describes how there is a potential conflict between fault-ride through and loss-of-mains detection. The frequency range and time windows for fault ride-through during frequency excursions

271

can be deduced by entries within G59/1 (ENA, 1991) (see Table 2-9) and IEEE 1574 (IEEE, 2003) (see Table 6-2). These are specified as required trip times for excursions of system frequency beyond the quoted ranges, but can also be interpreted as times for which ride-through is desirable so long as frequency remains within the non-trip range. It should be noted that the ride-through capability built into the frequency (and thus ROCOF) measurement algorithm of chapter 5 will provide good ride-through capability even without modification to the POR algorithm presented here.

DR size	Frequency range (Hz)	Clearing time(s)
≤ 30 kW	> 60.5	0.16
	< 59.3	0.16
> 30 kW	> 60.5	0.16
	< {59.8 – 57.0} (adjustable set point)	
	< 57.0	0.16

Table 6-2 : Under/Overfrequency clearing times required under IEEE 1547 (2003)

Vieira (2006b) proposes that this conflict be resolved by modifying settings of existing ROCOF relay designs so that fault ride-through is generally achievable, as the ROCOF relays will tend not to trip due to normal loss-of-mains conditions until the local frequency goes outside the ranges of Table 6-2. A better, simpler alternative might be to insert a "no-trip" frequency range into the loss-of-mains relay which would reset the relay (and ban trips) if frequency was almost exactly nominal. Whether this range would ideally be 0.99pu to 1.01pu (59.5 Hz to 60.5 Hz for a 60Hz system as referred to by Vieira (2006b)) or some other range (probably smaller) could be the subject of future research.

6.1.2 Appropriate settings for deployment within the national grid, and in the tests presented

The relay settings used for the tests which follow in section 6.1.3 are:-

- ROCOF trigger level $R_{Trigger}$ = 0.15 Hz/s, with a 5-cycle trigger. Thus, the phase offset which must be exceeded over 5 cycles continuous rolling integration, is 0.54° by (6.4)

- A Phase Offset trip threshold of Φ_{Trip} = 20° during normal conditions. The normal setting of 20° is determined by the desire to trip off the generator before there is a risk of excessive re-synchronisation currents (see Table 6-1).

- A Phase Offset trip threshold of 100° during faults (balanced faults <0.8pu or

unbalanced faults with >10% unbalance), which is decreased back to the normal setting over 3 seconds after the fault is no longer perceived. This profile is determined by analysing the worst-case results obtained from the simulations of Dysko (2006). Note that time-hysteresis (via the parameters "On_time" and "Off_time" in Figure 6-2) is applied to stop transient spikes/dips causing erroneous detection of faults.

The widening of the trip setting to 100° during faults introduces a small (but finite) chance that a 100° out-of-phase might occur due to auto-reclose action. It should be borne in mind, however, that for this to occur, an entirely unlikely chain of events must occur. An upstream breaker would first have to open due to the fault current flowing. This would cause a genuine loss-of-mains event, coupled potentially with a fault persisting within the un-intentionally islanded power system. The protection schemes within the islanded power system would almost certainly trip quickly in this scenario. If the fault persisted locally, the trip would likely be under-voltage, over-frequency or unbalance. If the fault was cleared locally, then one of several possibilities exist:-

- That the voltage will remain nearly nominal or rise due to reactive power outputs of local generation exceeding that of local load demand. In this case the trip threshold for the POR would revert to 20° and the island should be detected by this relay, or by over-voltage or over/under-frequency alarms.

- That the voltage will fall. This might lead to the 100° trip threshold for the POR being retained for some time, as an under-voltage event would be perceived as a fault by the algorithm presented in this thesis. Note, however, that a sustained under-voltage for more than about 0.5 second will lead to the local power system (or at least generator) being tripped by the guidelines of G59 (Table 2-9). IEEE 1547 only allows 160ms before such a trip will occur (Table 2-8).

Only if none of the above local trips occurs, and the undervoltage persists, and an auto-reclose action occurs, is a 100° out-of-phase synchronisation risked. The reclose action would have to occur within 0.5 seconds of the initial fault inception for the relay to still be using a 100° trip setting, and this is much shorter than standard practice for re-close action (Areva T&D, 2007). The wider POR trip setting is gradually reduced over 3 seconds after a fault condition is cleared. During this time, should a reclose action occur, there is some risk of a less-than-100° but more than 20° out-of-phase re-synchronisation occurring. Assuming re-close happens no sooner than 2 seconds the relay setting will have diminished to 46°.

It should also be re-stressed that some existing ROCOF and vector-shift relays disable themselves completely during faults (Dysko (2007), Freitas (2005) & Vieira (2006c)), whereas the solution proposed here is only a widening of thresholds and is thus less likely to allow out-of-phase resynchronisation.

6.1.3 Analysis of relay performance

Within the report by Dysko (2006), simulations are performed which test both the sensitivity and discrimination of candidate LOM detection algorithms. Simulations were performed using the UKGDS EHV network 1 (SEDG, 2008), and the most challenging scenarios involved distributed generators of the synchronous variety. The schematic for UKGDS EHV network 1 is reproduced in Figure 6-4.
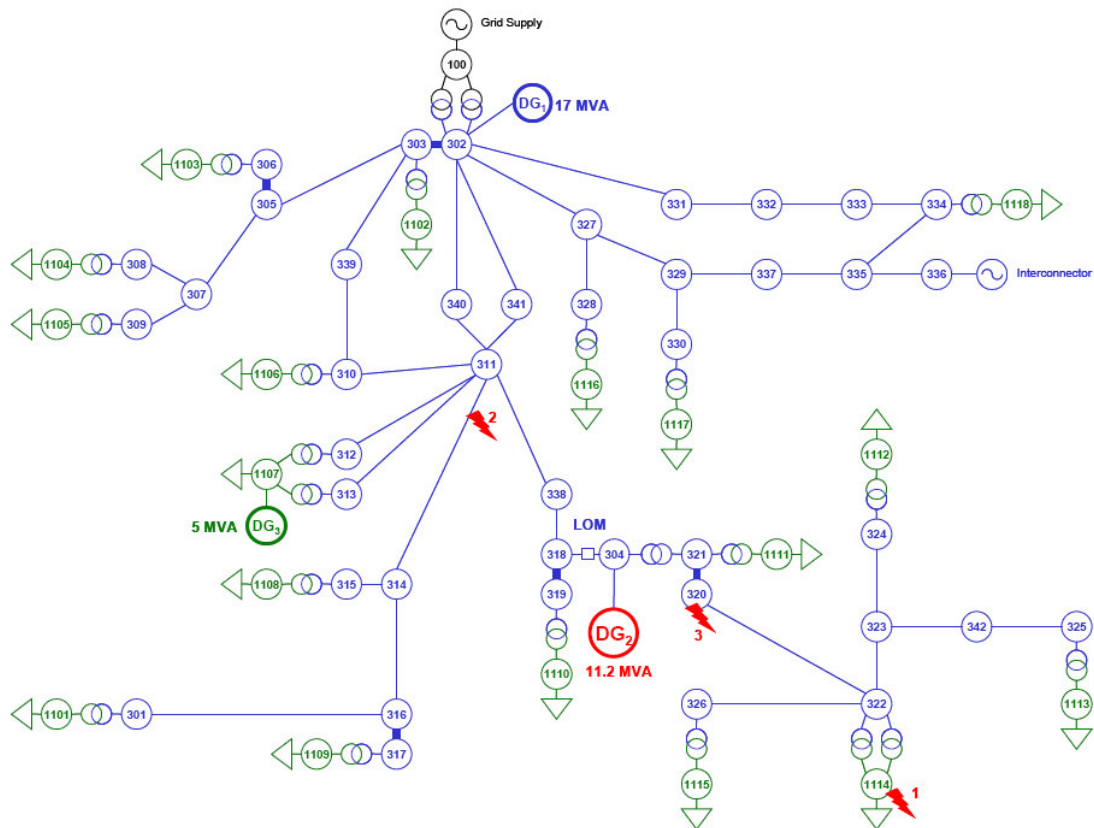
Figure 6-4 : UKGDS EHV network 1, reproduced from Dysko (2006)

From the results of Dysko (2006), the most challenging four scenarios are:-

- Sensitivity; detection of LOM with generator power output set 2.5% higher than local load power. (Scenario 3 in Dysko (2006)).

- Sensitivity; detection of LOM with generator power output set 2.5% lower than local load power. (Scenario 5 in Dysko (2006)).

- Discrimination; avoidance of trip during/following a 3-phase fault for 200ms at

position 2 (upstream of the distributed generator, within the distribution network). (Scenario 11 in Dysko (2006))

- Discrimination; avoidance of trip during/following a 3-phase fault for 200ms at position 3 (close downstream of the distributed generator). (Scenario 14 in Dysko (2006))

When detecting a genuine loss-of-mains (LOM) event, the detection will not be instant but will occur some time after the event begins. As described above, the purpose of the proposed POR is to create a trip signal when the local generator becomes out-of-phase with the parent network by an amount which would cause undesirable currents and torques should an un-controlled reconnection subsequently occur, for example by action of an auto-reclose relay. By (6.3), the trip time with a trip threshold of 20° might be as small as 100ms for a 10 Hz/s event or bigger than 860ms for an event which only just breaches a 0.15 Hz/s ROCOF trigger threshold.

It is interesting to compare these times with those required by IEEE 1547 (IEEE, 2003), G59/1 (ENA, 1991) and ETR 113 (ENA, 1995) the documents describing distributed generation connection in the UK and the USA:-

- IEEE 1547 states that the island must be detected within 2 seconds.

- G59 simply states that LOM relay settings should be "agreed" with the PES (DNO). G59 does say that -6% (<47Hz) or +1% (>50.5Hz) events should cause trips within 0.5s, but this does not have much relevance for loss-of-mains since several seconds may elapse between the loss-of-mains event and a breach of these frequency thresholds; i.e. the actual detection time for loss-of-mains could be 0.5s plus several seconds, and still meet the G59/1 recommendations.

- ETR 113 states than detection within 1 second may be required in some UK rural distribution networks due to potential auto-reclose times in feeders, but this is not a general requirement across the UK.

The approach taken by this thesis (and the relay proposed here) is that the tripping should take as long as it needs, so long as the risk to generator and distribution equipment is minimised, and that the trip does actually occur within the prescribed 1-2 seconds. The electrocution risk to personnel in touching parts of "presumed dead", but actually islanded, systems, is small within 1-2 seconds of any deliberate disconnection for maintenance purposes. For the tests performed in the subsequent sections, the 2-second detection requirement of IEEE 1547 (IEEE, 2003) has been used. The recommendation of ETR 113 is for a 1 second detection time. This can be achieved for the same non-detection

275

zone (NDZ) by tightening the trip threshold, or the trip threshold can be left the same but the NDZ will be larger.

To test the sensitivity and discrimination of the proposed new relay, the archived simulation results (3-phase voltage waveforms) for these 4 scenarios from Dysko (2006) are injected into simulations of the relay. The front-end of the relay is the frequency measurement algorithm described in chapter 5, and this feeds into the POR algorithm shown in Figure 6-1 and Figure 6-2. The sample rate for all the major algorithms is 500 Sa/s. An important characteristic of the relay is its rejection of interference due to noise and harmonics. However, the archived simulation results for the scenarios contains very pure waveforms of almost 0% THD (apart from during the fault inception/clearance), due to the limits of the simulation. To fully test the algorithms an intermediate pre-processing stage has been developed for this analysis. The archived simulation waveforms are analysed for frequency/magnitude/phase by the algorithms of chapters 3 and 5. Harmonic content (2nd to 40th) is then added to the original archived waveform with appropriate phases to synthesise the 2nd-worst case microgrid waveform of section 2.7.2,

Table 2-6 and Figure 2-5.
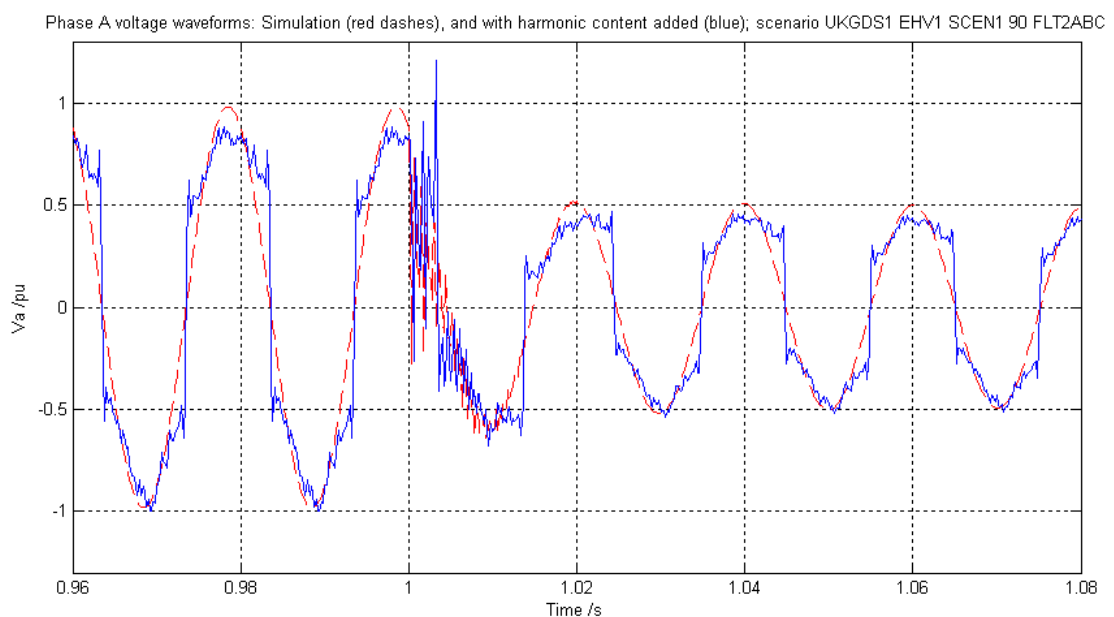
Figure 6-5 : Addition of harmonic content to the simulated waveforms; phase A voltage waveform from scenario simulation (red dashes), and with harmonic content added (blue solid line).

This produces a three-phase voltage waveform set with the same frequency, amplitudes

276

and phases as the simulation, but with a THD$_V$ of approximately 28%. These corrupted signals are then used as the voltage inputs to the relay algorithms. An example of this process is shown in Figure 6-5, which is an excerpt from scenario 11 of Dysko (2006), with a 200ms fault beginning at t=1 second.

In addition to harmonic content, the analogue low-pass anti-aliasing filter, instrumentation noise and ADC quantisation/oversampling/downsampling are also simulated as described in chapter 3, section 3.4.

6.1.3.1 Sensitivity testing; detection of LOM with generator power output set 2.5% higher than local load power demand

This test uses scenario 3 from Dysko (2006). At t=1 second, the circuit breaker labelled "LOM" in Figure 6-4 is opened to create a loss-of-mains event. The relay should detect the LOM condition within 2 seconds. Figure 6-6 to Figure 6-9 show the results of this test. The relay successfully throws a trip signal, 910ms after the LOM event occurred. Triggering occurred 180ms after the LOM event, and at this time the 0.54 degrees which the 5-cycle phase offset trigger has accumulated is pre-loaded into the main phase offset integrator. This shows as a small initial brief rise in Figure 6-9 at t=1.18 seconds. The 20 degree threshold is exceeded 910ms after the LOM event, at t=1.91 seconds.

Thus, the relay passes this test by both the IEEE 1547 (IEEE, 2003) requirement for a 2 second trip and also by the ETR 113 (ENA, 1995) requirement for a 1-second trip. Notably, a ROCOF relay set to a familiar setting of 0.15 or 0.2Hz/s for 200ms may not trip in this scenario, as shown by Figure 6-7. The ROCOF threshold and/or time qualification setting could be lowered, but note that noise would then become a serious issue and probably lead to spurious tripping. Comparing Figure 6-7 and Figure 6-9 shows how much better the event is detected by the POR than by a ROCOF relay.

Also of note, is that here a phase offset threshold of 20° was small enough to detect the event, whereas in Dysko (2006), for this scenario a setting as low as 5° was required. This discrepancy is because here the relay has been allowed longer to detect the trip, up to 2 seconds as per IEEE 1547 (IEEE, 2003) , whereas Dysko (2006) applied a requirement for a trip within 500ms, conservatively half of the ETR 113 (ENA, 1995) 1-second requirement. For a trip within 500ms, Figure 6-9 indeed shows that a setting of about 5° would be required. This shows that the POR algorithms presented here are equal in sensitivity to the algorithm used in Dysko (2006).
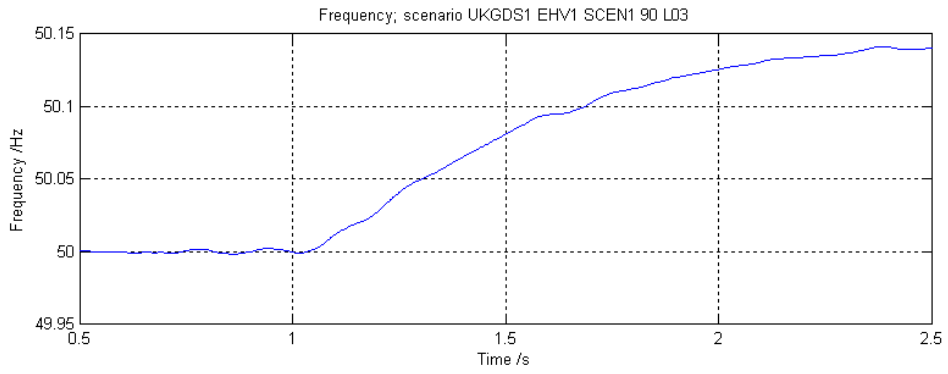
Figure 6-6 : Scenario 3; +2.5% power; Frequency excursion
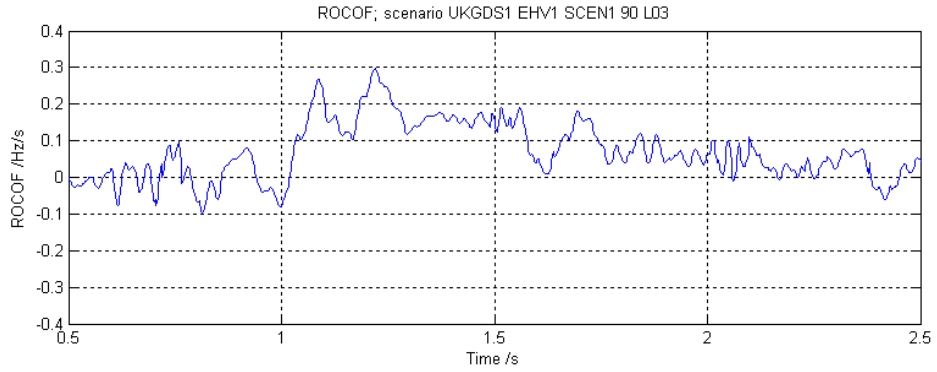
Figure 6-7 : Scenario 3; +2.5% power; ROCOF
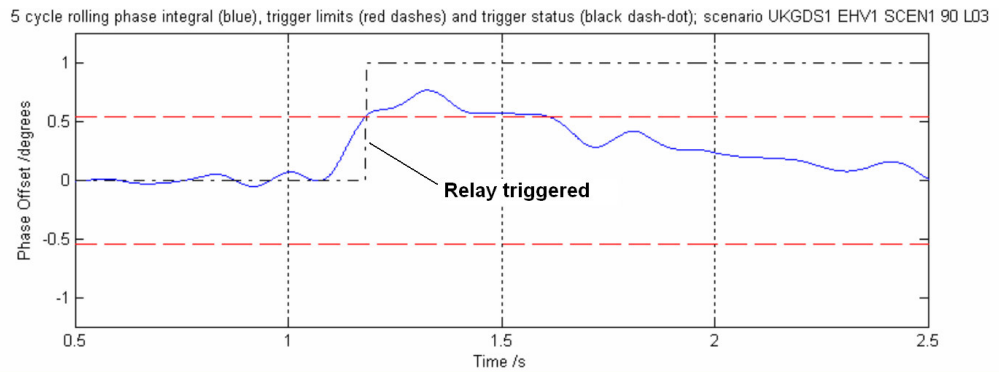
Figure 6-8 : Scenario 3; +2.5% power; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)
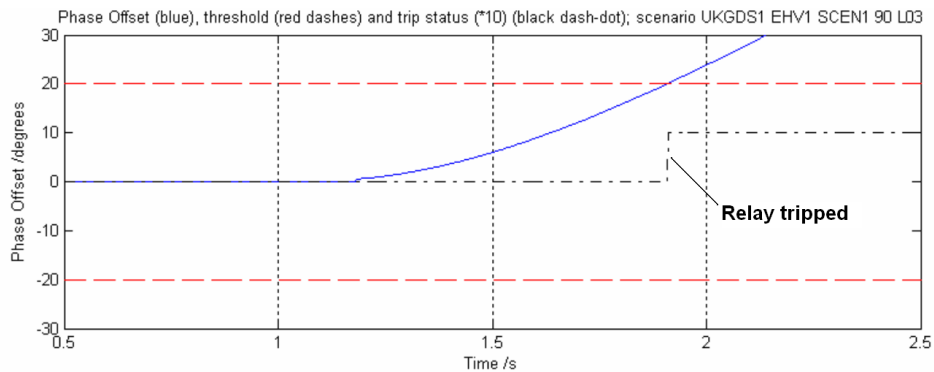
Figure 6-9 : Scenario 3; +2.5% power; Phase Offset (blue, with threshold in dashed red) and Trip signal (black dash-dot)

278

6.1.3.2 Sensitivity testing; detection of LOM with generator power output set 2.5% lower than local load power demand

This test uses scenario 5 from Dysko (2006). At t=1 second, the circuit breaker labelled "LOM" in Figure 6-4 is opened to create a loss-of-mains event. The relay should detect the LOM condition within 2 seconds.

Again, the test is successful. This time the trip takes slightly longer, at 1120ms, which again meets the IEEE 1547 (IEEE, 2003) requirement for a 2 second trip.
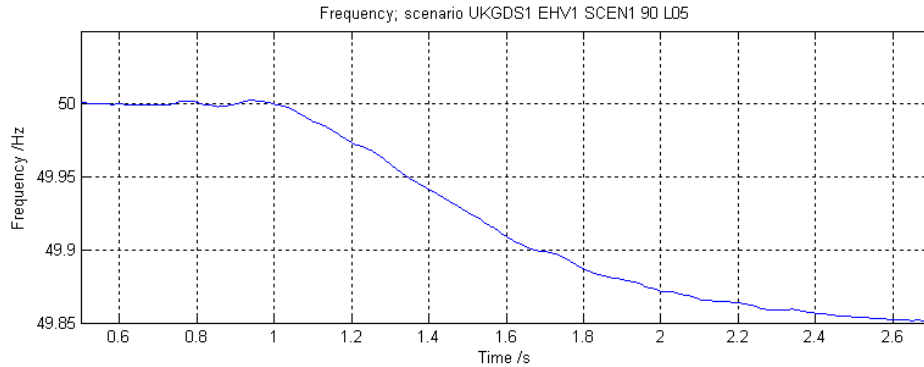
Figure 6-10 : Scenario 5; -2.5% power; Frequency excursion
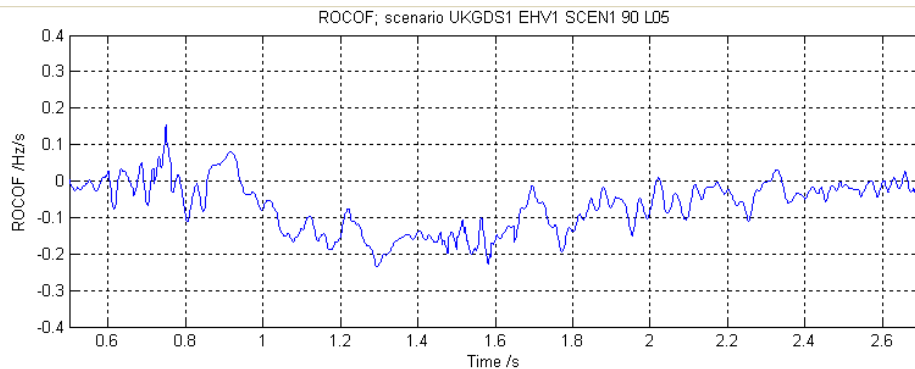
Figure 6-11 : Scenario 5; -2.5% power; ROCOF
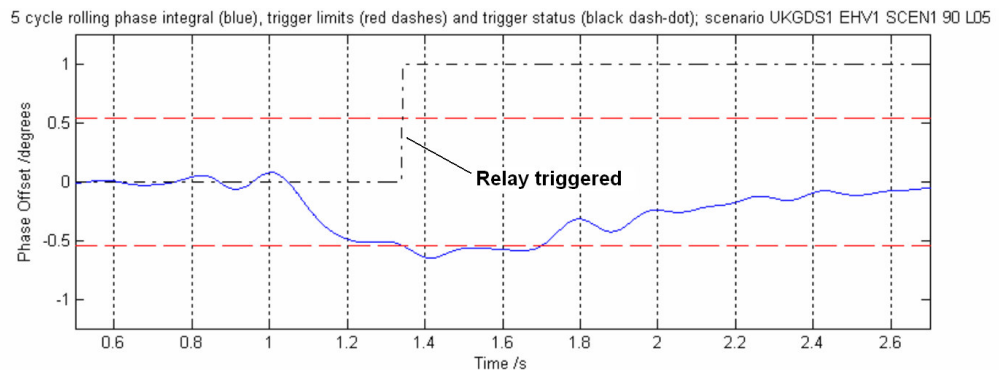
Figure 6-12 : Scenario 5; -2.5% power; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)
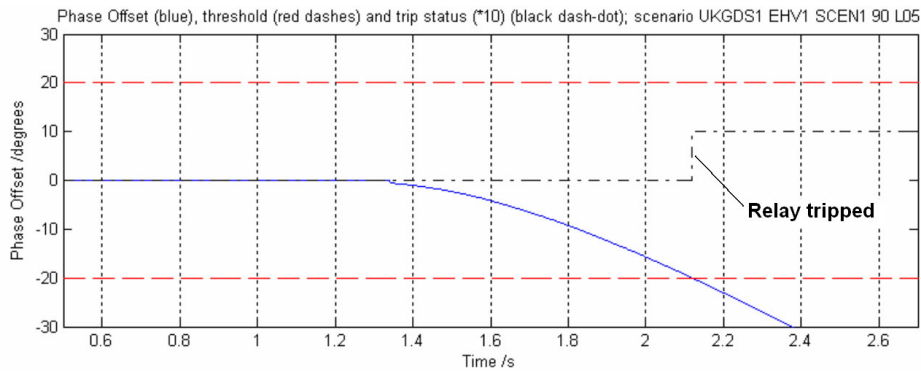
Figure 6-13 : Scenario 5; -2.5% power; Phase Offset (blue, with threshold in dashed red) and Trip signal (black dash-dot)

6.1.3.3 Discrimination testing; avoidance of spurious trip due to 200ms 3-phase upstream fault

This test uses scenario 11 from Dysko (2006). At t=1 second, a three-phase fault occurs at position 2 of in Figure 6-4, i.e. node 311. The relay should ideally not give a LOM trip for this scenario.

The fault begins at t=1s. During the fault itself, the frequency variation is small, (although voltages are depressed). When the fault clears, at t=1.2s, a significant power system oscillation occurs, which takes several seconds to decay. Frequency (and hence ROCOF) excursions are large, but oscillatory, during this time. The fault is detected (via low positive sequence) shortly after t=1s, and the phase offset trip threshold is widened automatically to 100° (Figure 6-17). Shortly thereafter, the 0.54° triggering threshold is violated by the rolling 5-cycle phase offset (Figure 6-16). However, due to the oscillatory nature of the frequency excursion, the triggering is continually reset by the zero crossings registered by the rolling 5-cycle phase offset measurement. These resets occur shortly after the peaks (positive and negative) of the frequency excursions. The final phase offset integral never breaches the trip threshold, which decreases back to the normal 20°, 3 seconds after the fault clearance. No LOM trip signal is given.

Note that the raw ROCOF data in Figure 6-15 shows significant spikes, because it is calculated by a two-sample differentiation from the frequency data. The frequency data has a slightly non-continuous nature due to actions of seeding algorithms and ripple-rejection filters during the most dynamic events. These algorithms and filters are described in chapters 3 and 5. The spiky/noisy ROCOF data does not cause a problem for the POR proposed here, which is specifically designed to cope with this kind of effect. The use of ROCOF data in this relay is merely as an intermediate stage to remove DC offset from the "Frequency Offset" result, and at the same time to provide an indication of

dynamic performance for graphical presentation. Smoother ROCOF data can be created, if required for cosmetic or technical purposes, by differentiating frequency over longer timeframes, or by averaging the two-sample differentiated values over a number of samples (these two methods being mathematically identical).

A ROCOF relay set to 0.2Hz/s for 200ms, would trip during this event, unless it was automatically disabled (or the ROCOF trip threshold widened to >10 Hz/s) during and immediately subsequent to the fault.
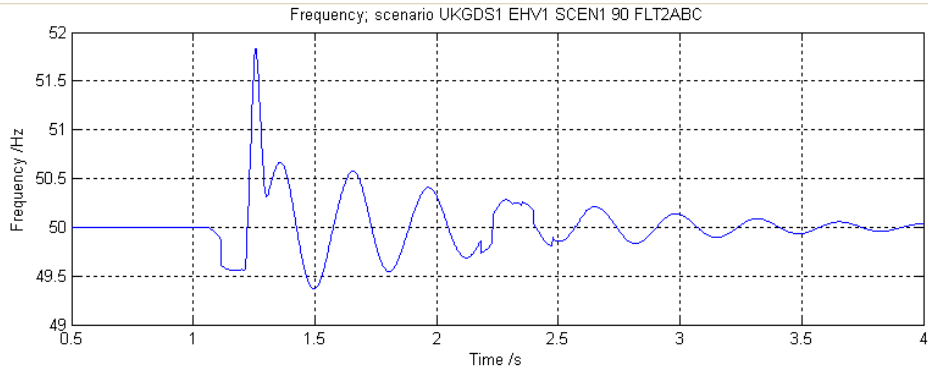
Figure 6-14 : Scenario 11; 3-phase fault at position 2; Frequency excursion
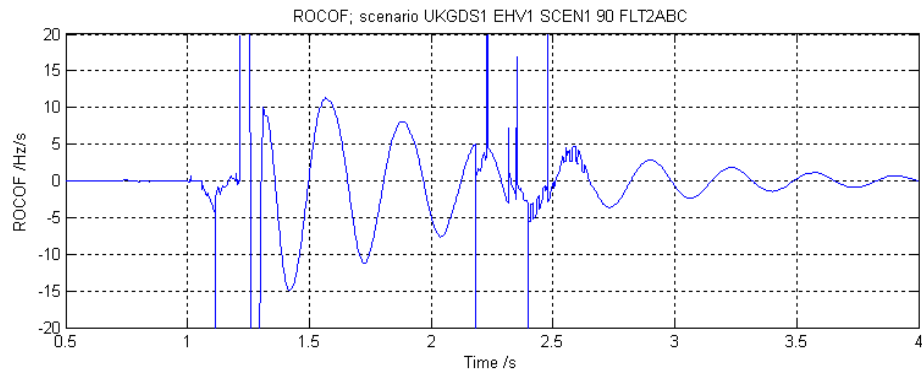
Figure 6-15 : Scenario 11; 3-phase fault at position 2; ROCOF
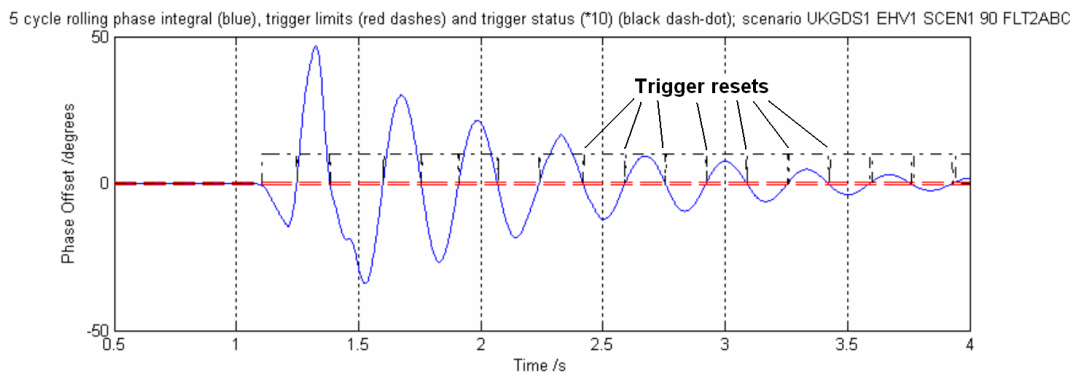
Figure 6-16 : Scenario 11; 3-phase fault at position 2; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)
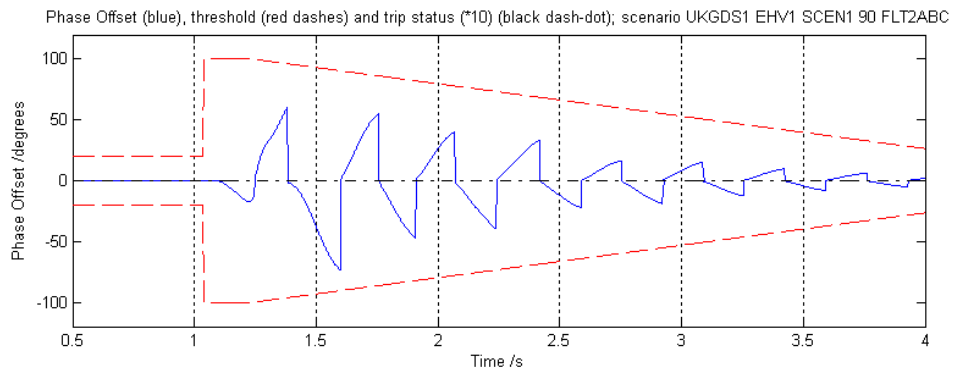
Figure 6-17 : Scenario 11; 3-phase fault at position 2; Phase Offset (blue, with threshold in dashed red)

6.1.3.4 Discrimination testing; avoidance of spurious trip due to 200ms close-in downstream 3-phase fault

This test uses scenario 14 from Dysko (2006). At t=1 second, a three-phase fault occurs at position 3 in Figure 6-4, i.e. node 320. The relay should ideally not give a LOM trip for this scenario. This turns out to be the case. The comments and observations from section 6.1.3.3 all apply equally to this scenario.
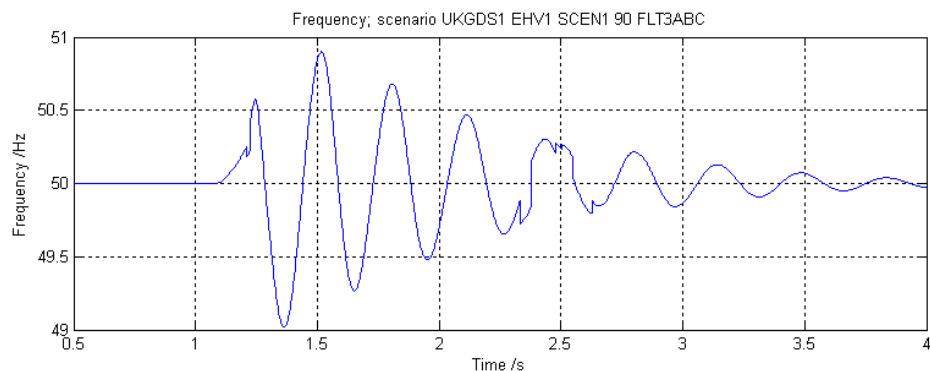
Figure 6-18 : Scenario 11; 3-phase fault at position 3; Frequency excursion
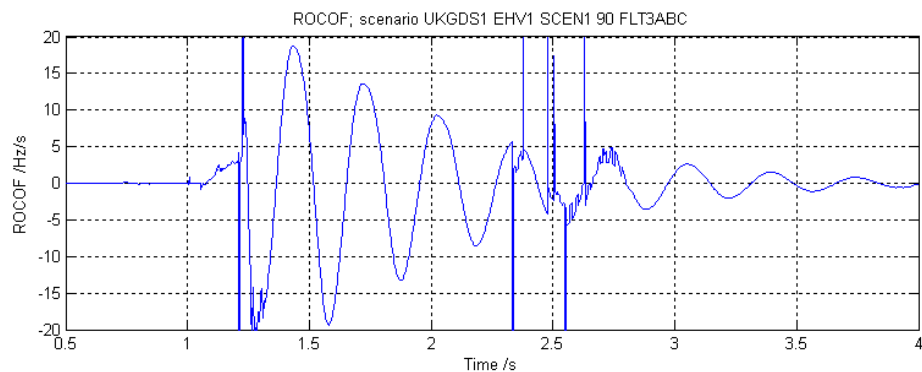
Figure 6-19 : Scenario 11; 3-phase fault at position 3; ROCOF
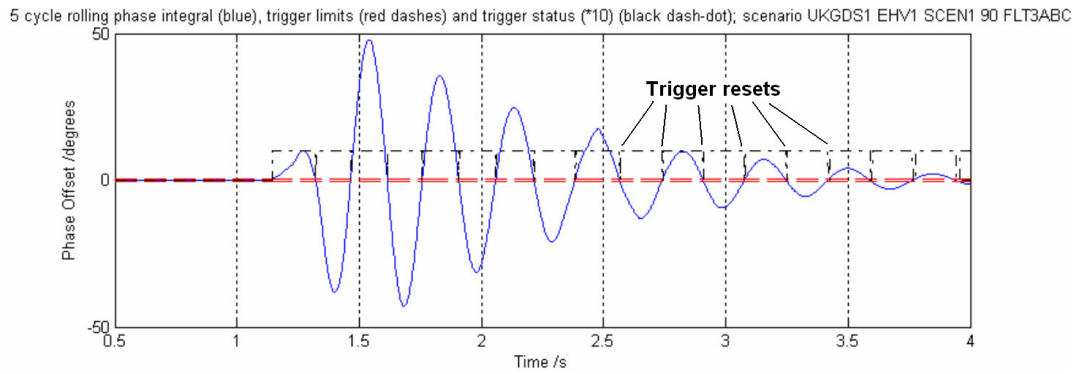
Figure 6-20 : Scenario 11; 3-phase fault at position 3; Rolling 5-cycle phase integral (blue, with threshold in dashed red) and triggering (black dash-dot)
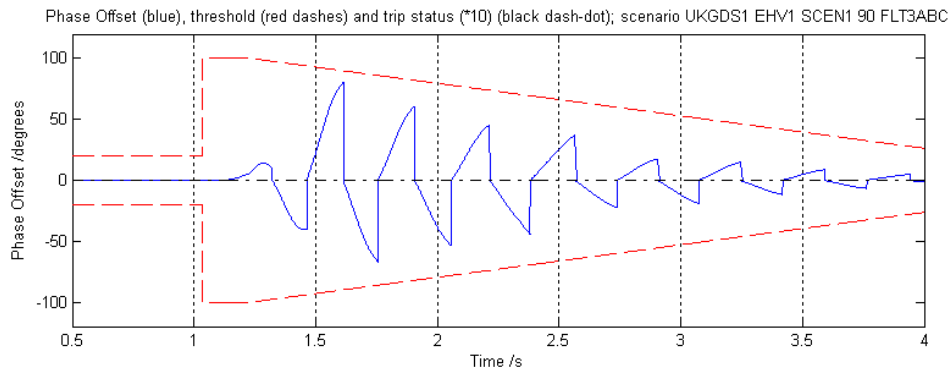
Figure 6-21 : Scenario 11; 3-phase fault at position 3; Phase Offset (blue, with threshold in dashed red)

6.1.4   Further comments on the performance of the POR

The results from Dysko (2006), suggested that a setting of about 45° would be enough to avoid trips during the fault scenarios, whereas the results here (Figure 6-17 and Figure 6-21) show that about 75° would be required, if the setting was fixed. This discrepancy is almost certainly due to the difference in the triggering methods used. In Dysko (2006), the proposed relay does not have a triggering subsystem as such, but uses a high-pass filter to continually attenuate the accumulated phase offsets. This stops the relay tripping during normal network operation, but will also tend to reduce the accumulated phase offset even during relatively short events. In the relay proposed in this thesis, once the triggering subsystem fires, the overall phase offset integral accumulates without attenuation, until it either trips or is reset by the trigger subsystem. Thus, the relay proposed in this thesis provides a more accurate measure of the actual accumulated phase offset once triggering starts, and can easily accumulate higher values of accumulated phase than the relay proposed by Dysko (2006). The cut-off frequency and design of the high-pass filter in Dysko (2006) would have to be known in detail to fully compare the two algorithms. It should be noted that the thresholds for the triggering subsystem (and for the Dysko relay the design of the high-pass filter), are equally important as the trip thresholds, in terms of setting the relay for optimum sensitivity and discrimination.

283

The results presented above show good sensitivity and discrimination. This relay could be deployed at distributed generator sites, where the generators are connected to a large power system, with the settings proposed here. Its performance should, according to the analysis performed to date, significantly surpass that of conventional ROCOF and Vector Shift relays. The 2 sensitivity tests and 2 discrimination tests presented in the previous sections show that a ROCOF relay set to 0.2Hz/s with a 200ms qualification time would not have tripped for the 2 genuine LOM events, but would have tripped (with ROCOF reaching >10Hz/s) for both the non-LOM events. By comparison, the proposed POR relay successfully detects and discriminates the relevant scenarios. It might be argued that:-

1. The ROCOF relay settings could be tightened, to enable greater sensitivity. However, the noise level of ROCOF is approximately 0.1 Hz/s, even using the frequency measurement algorithm presented in this thesis. Thus, any ROCOF trip setting lower than 0.2 Hz/s risks spurious tripping due to noise unless the qualification time for tripping is increased (which increases the minimum possible trip time) or the ROCOF values are averaged/filtered (which increases trip time).

2. The ROCOF relay trip could be disabled during faults (as previously described). However, this does not retain any level of LOM protection during the time which the disable action occurs.

However, a further challenge for LOM detection remains to be addressed. The natural frequency disturbances in microgrids are much larger than those of a power system such as the UK national grid. The expected rates of change of frequency were predicted in section 2.1, and summarised in Table 2-1. Notably, within a 100kVA microgrid, even switching on a single kettle at 3kW can be enough to cause a brief 0.4 Hz/s frequency deviation. Larger load steps can cause larger disturbances. The problem scenario is thus that a very small power system, say of 10kVA rating, is "grid-connected" to a parent network of only 100kVA. In this case, the 10kVA microgrid (generator plus its local load) will measure ROCOF events regularly greater than 0.4 Hz/s. Governor actions will act to mitigate these deviations within the constrains of the control system dynamics, but with a trigger setting of 0.15Hz/s, there is a significant risk that the relay (with the settings used above) would

1) trigger, as ROCOF surpasses 0.15Hz/s

2) trip, if the frequency did not stabilise to a new level quickly enough to avoid a 20° phase offset being accumulated. Note that if frequency stabilises, it causes ROCOF to become about 0 (with noise), which causes the 5-cycle phase offset rolling integral to also become about 0 (with noise), and hence cross zero to reset the trigger.

A potential solution to avoiding such spurious trips when the parent network is of limited size, without employing communication systems or active LOM-detection techniques, is to widen the ROCOF trigger to avoid spurious trips. This would be a better approach than widening the trip threshold of 20°, since this 20° figure relates to the worst uncontrolled synchronisation angle which would ideally be risked.

It might be possible to design an algorithm which continually monitors system frequency deviations and autonomously sets a suitable ROCOF trigger threshold. Or, knowledge of network configuration might be used to deliberately adjust the relay settings when the parent network itself joins (or leaves) a much larger power system which brings greater frequency stability. A combination of the above two possibilities might also be realised.

The impact of widening the trigger threshold would be two-fold:

- Larger system frequency deviations, due to larger load switching and generator despatching (as per-unit proportions of the total system inertia), could be accommodated without the relay triggering and risking a spurious trip.

- During genuine LOM events where active and reactive load powers are well matched to the generator(s) output power set-points, the risk of non-detection is raised. This can be seen by imagining a ROCOF trigger threshold of 0.4Hz/s in section 6.1.3.1 or 6.1.3.2 above. In those scenarios, with a ROCOF trigger threshold of 0.15Hz/s (converted to a 5-cycle phase offset of 0.54°), the relay only just triggered. If the trigger threshold was doubled, the relay would neither trigger nor trip.

Within a microgrid context, therefore, the relay may need to be de-sensitised to avoid constant spurious tripping. One option to reduce the risk of subsequent non-detection of genuine LOM events is to manage the generators and/or network topology such that there is a very low risk of any generator P/Q power output set-point being a close match for any local load P/Q combination. Only a small imbalance in either P (real power) or Q (reactive power), and not necessarily both, is required to cause imbalance in an unintentionally islanded power system which will allow detection of the LOM event. This effect, along with a novel algorithm to ensure such an imbalance, is presented in section 6.2 to follow.

6.1.5   Summary and further work opportunities arising from this section

- A novel algorithm for a Phase Offset Relay (POR), for the detection of loss-of-mains, has been designed and coded. The algorithm is fully robust for real-time

implementation and is significantly more immune to noise and interference effects than a ROCOF relay, due to its use of Phase as the key measurement parameter. Phase is calculated as the double-integral of ROCOF which is equivalent to two stages of averaging, thus giving good rejection of noise on a frequency/ROCOF measurement.

- The POR allows for variable trip times, graded so that the important parameter is the potential phase angle relative to the parent network which would result due to an uncontrolled out-of-phase resynchronisation.

- Both the triggering algorithm setting and the trip setting need to be appropriately set to provide the best balance of discrimination and sensitivity.

- Good discrimination can be provided for ride-through of faults by using a small algorithm which temporarily allows larger phase offset trip settings during faults (which can be detected using positive sequence and unbalance thresholds applied to 3-phase voltage measurements) The trip settings can be tapered from the wider settings back to normal (20°) over 3 seconds after a fault is cleared, to allow for network oscillations to die away.

- The network simulation results suggest that using a trigger setting of 0.15-0.2 Hz/s and a normal trip setting of 20° allows detection of loss-of-mains in less than 1.5 seconds; assuming that the unintentional power islands are more than 2.5% mismatched in terms of active power. These settings could be used sensibly for grid-connected systems attached to the UK national grid.

- When the parent network is smaller, the trigger setting of 0.2 Hz/s may be too small, allowing numerous spurious trips. In this case the trigger setting must be widened, which de-sensitises the relay. This is explored further in section 6.2.

Further work opportunities include:-

- Design and test an algorithm to automatically adapt the ROCOF triggering level $R_{Trigger}$ for the proposed relay, for different scenarios/size of parent network. The algorithm would need to monitor frequency deviations due to "normal" network behaviour.

6.2 Application to loss-of-mains detection; discrimination testing using real-world transient data

The most obvious case of spurious loss-of-mains detections is due to genuine power system frequency disturbances which cause ROCOF (rate of change of frequency) to exceed trigger or trip settings within loss-of-mains relays. Likely values of ROCOF within different sized microgrids, from 100kVA to 60GVA, were analysed in section 2.1 and Table 2-1. For distributed generators connected to a distribution system in the UK, the size of the entire

power system is approximately 60GVA. According to Table 2-1, ROCOF should be limited to about 0.01 Hz/s for normal operation, and about 0.1-0.2 Hz/s for worst-case power station tripping leading to a ~2GW step change in generation.

To verify that the estimated value of 0.01 Hz/s from Table 2-1 is correct, a data logging application was created. This executes on the ADI RTS (Real-Time-Station) (ADI, 2008) with a sample rate of 2000 Sa/s. The sampled values are three-phase voltages at the LV side of an 11kV/433V transformer, shown on in Figure 6-28, which is part of the laboratory infrastructure. The application software processes the data in real-time using the algorithms produced during this thesis. Parameters monitored include frequency, ROCOF, fundamental voltage amplitudes, unbalance, THD and possible spurious loss-of-mains detection events. Data is logged every second, with the most interesting event causing additional capture at the full 2000 Sa/s frame rate for a 60 second window (30 seconds pre-trigger plus 30 seconds post-trigger data).

Over 825 hours of monitoring (summer and winter), were logged during 2007. All the genuine ROCOF events captured were restricted to ROCOF rates of less than 0.025 Hz/s, on a second-by-second basis. This suggests that the analysis of Table 2-1 is correct (to within a power of 2). The term "genuine" is here used to restrict the analysis to events which appear as genuine frequency slides which occur over several seconds.

In contrast, 7 events were logged which transiently show much larger rates of change of frequency. These only show up on the 1-second logged data as spikes. The thresholds for detecting "interesting" events within the logging application can be set to very low levels, much tighter than a normal protection relay would be set. In this way, such interesting deviations from static conditions can be logged at 2000 Sa/s for further analysis.

Interestingly, of the 825 hours data which was logged, 5 of the 7 transient events which were not "genuine" rates-of-change of frequency, but were transient events, occurred within a single 6 hour period. The remaining 2 transient events occurred 4 days later within a 15 minute period. Analysis of the events would suggest that they are distant unbalanced faults, switching or tap-changing events. The faults do not cause voltage depressions sufficient to cause widening of the POR trip settings as described in section 6.1.

The worst of the 7 transient events contains only a small (0.02pu) voltage step. The transient would appear to be due to either a load step or a tap-change within the

distribution network. A small phase change also results, which leads to a brief perception of frequency change. It is this perceived frequency change which presents the risk of spurious LOM detection. The measurement of ROCOF is as high as 2 Hz/s (both in the negative and positive directions). This is a large amount (much larger than the "genuine" background 0.01-0.025 Hz/s ROCOF rates). This finding, when added to the work of section 6.1, indicates that spurious trips of loss-of-mains relays based upon frequency, phase or ROCOF detection are far more likely to be caused by faults and transient events than genuine frequency changes.

To test the discrimination of the proposed POR against such transient events, the captured event waveforms (sampled phase voltages) from the laboratory hardware, can be replayed into the proposed relay code. The logged data is sampled at 2000 Sa/s, so to test the main algorithm at a sample rate of 500 Sa/s (nominally 10 samples per cycle at 50 Hz) the data is FIR-filtered using a 3-zero/3-pole filter (to remove potentially aliased harmonics as described in 4.5) and down-sampled by a factor of 4, before being input to the main algorithms. The main algorithms are the amplitude/phase/frequency measurements from chapters 3 & 5 and the POR from section 6.1.1.

The graphs below show the performance of the algorithms and the POR due to the worst of the 7 transients recorded. The settings for the POR are a trigger threshold $R_{Trigger}$ = 0.15 Hz/s and a trip threshold Φ_{Trip} = 20˚. These are the identical settings used for the testing in 6.1, and represent the tightest settings which are anticipated to be used in practice.
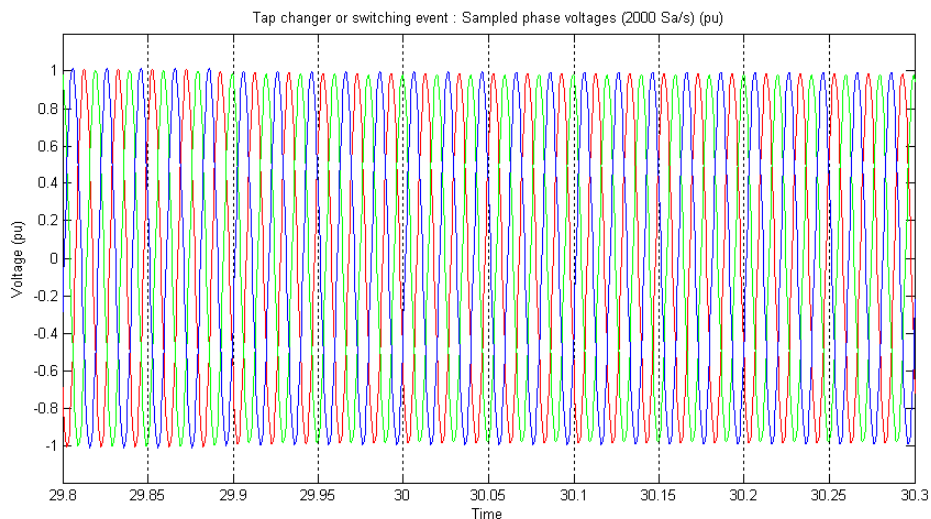
Figure 6-22 : Transient event recorded at local 11kV/433V transformer; sampled voltages (pu)

Clearly, from Figure 6-22, there is no significant voltage dip or surge on any phase.
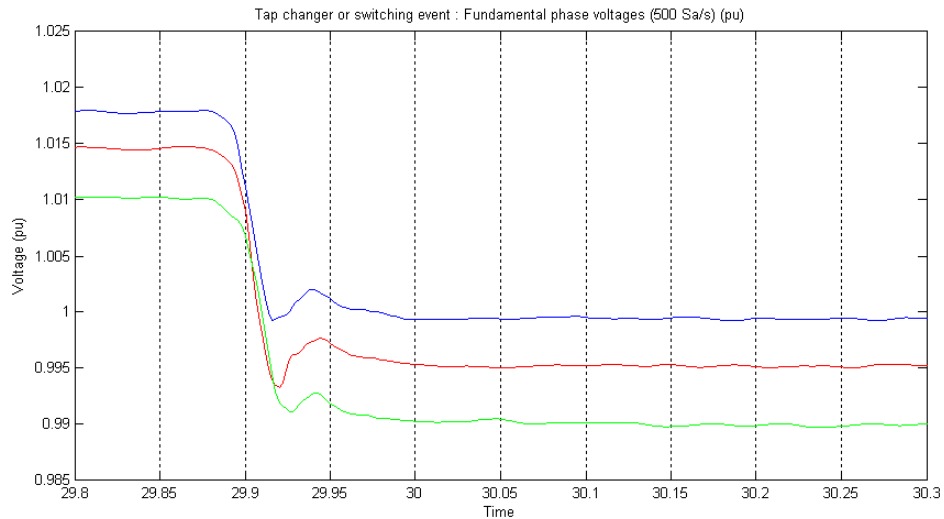
Figure 6-23 : Transient event recorded at local 11kV/433V transformer; fundamental voltages (pu)

Analysis of the fundamental voltage amplitudes shows that a 0.02pu voltage step event does in fact occur at t≈29.88s. This may be due to a tap changer or switching.
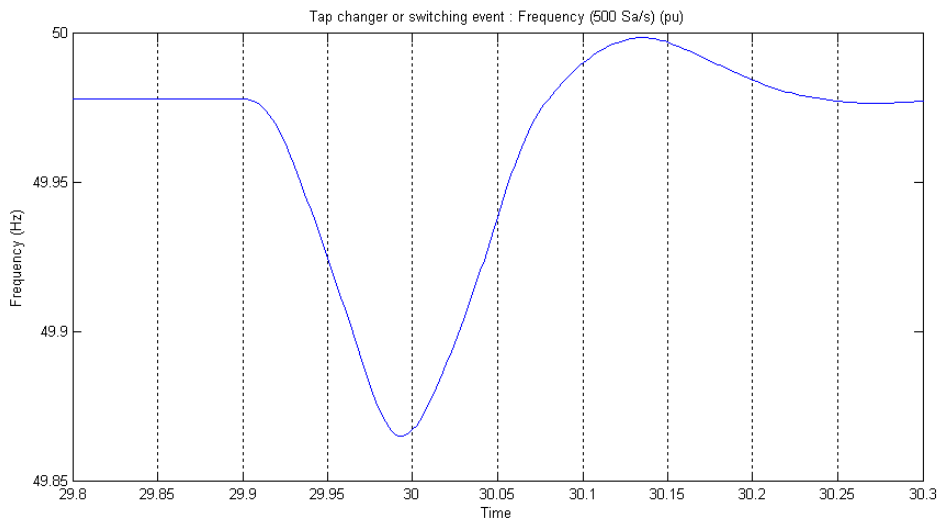
Figure 6-24 : Transient event recorded at local 11kV/433V transformer; frequency (Hz)

The measured frequency (Figure 6-24) shows a transient dip, lasting 100ms. This shape results due to a phase step in the voltages during the time period t=29.89 to t=29.9 seconds, and is smeared in time due to the response time of the FIR filters within the measurement algorithms (see chapters 3 & 5).

289
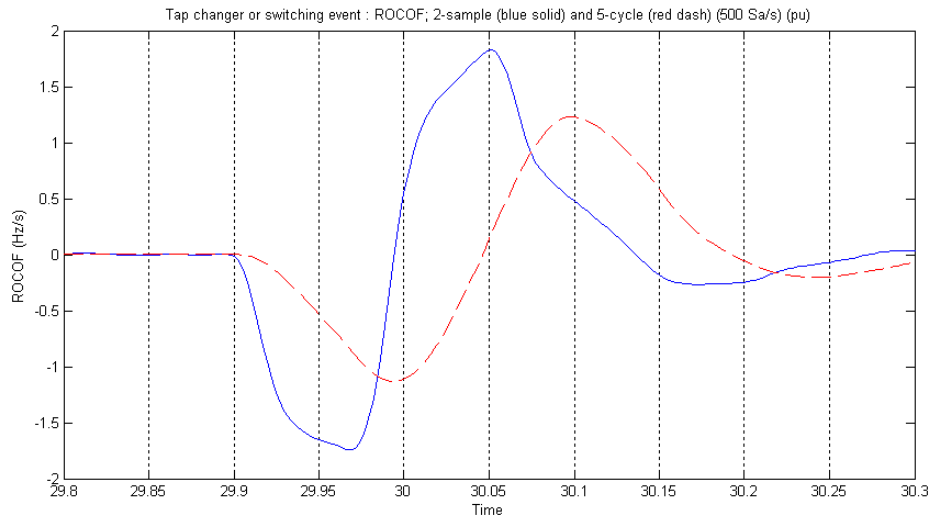
Figure 6-25 : Transient event recorded at local 11kV/433V transformer; ROCOF (Hz/s)

Figure 6-25 shows the calculated ROCOF. The POR uses the ROCOF value shown as a blue solid line, calculated by the 2-sample differentiation of the frequency shown in Figure 6-24. The red dashed line shows the same data averaged over a further 5 cycles (100ms); it is not used in the subsequent analysis but shows that even a ROCOF relay which smoothes/filters the ROCOF data will show a >1Hz/s deviation. Thus a ROCOF relay set to 0.5Hz/s with a qualification time of less than 100ms risks tripping.
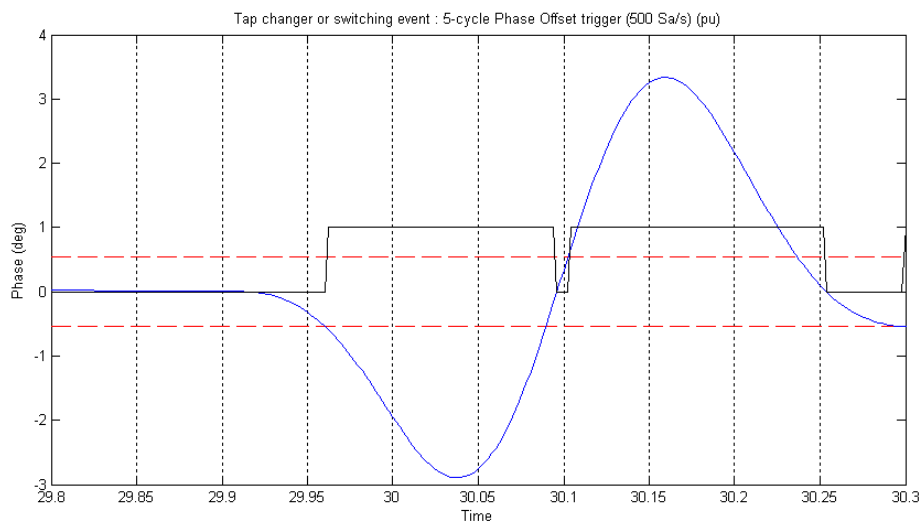
Figure 6-26 : Transient event recorded at local 11kV/433V transformer; POR triggering

The POR relay is triggered at t=29.97s when the rolling 5-cycle phase offset (blue solid line) exceeds the trip threshold (shown in red dashes) set by $R_{Trigger}$ = 0.15 Hz/s (Figure 6-26). The relay is reset at t=30.09s when the rolling 5-cycle phase offset crosses through zero, and is then triggered again at t=30.1s due to another violation of the threshold.
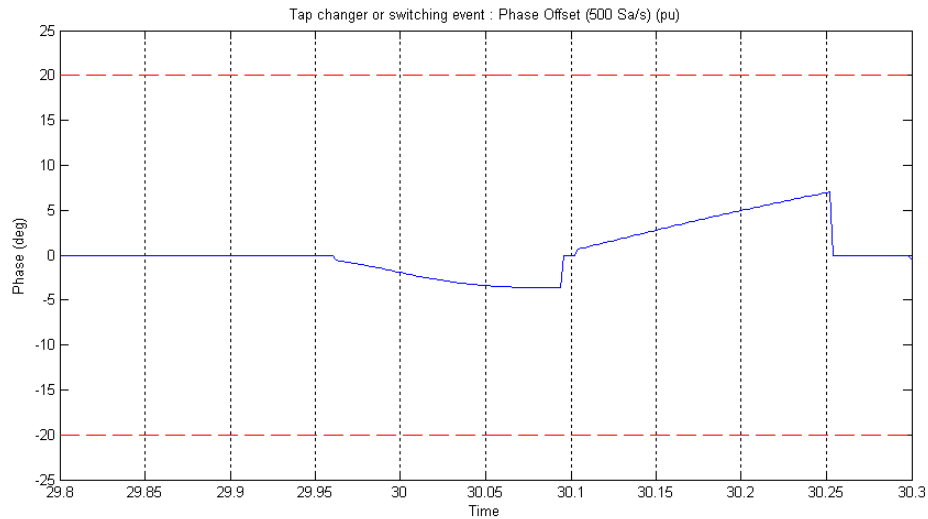
Figure 6-27 : Transient event recorded at local 11kV/433V transformer; phase offset

Although ROCOF is large (over 1Hz/s as shown in Figure 6-25), the duration for the event is short, and the ROCOF value reverses from a –ve to a +ve direction, causing the POR triggering to reset, as shown in Figure 6-26. The phase offset never reaches the trip threshold of Φ_{Trip} = 20°. The largest angle reached is about 7.1°. This would have caused a trip if a tighter trip setting of 5° had been used (as used by Dysko(2006)) to achieve 500ms tripping with a 2.5% active power unbalance (see section 6.1.3.1).

6.2.1   Summary findings and further work opportunities arising from discrimination testing using real-world transient data.

- Based upon 825 hours of logged data, the usual ROCOF rates observed on the UK national grid fall within ±0.025 Hz/s. This correlates with predictions of Table 2-1.

- Based upon the 825 hours of logged data, some of the worst events which can cause problems for spurious loss-of-mains detection are not the frequency slides due to generation/load imbalance, but load/voltage switching events which cause transient voltage phase changes. These are picked up by frequency measurement algorithms (see section 5.7.4) and appear as transient ROCOF events. Faults will also cause similar problems, as already analysed in section 6.1.3.

- The observed transient ROCOF rates due to switching events were as high as ±1.8 Hz/s.

- For the worst event observed, the proposed POR does not trip, despite the trigger threshold ($R_{Trigger}$ = 0.15 Hz/s) being substantially less than the peak ROCOF value of 1.8 Hz/s. The relay successfully discriminates between the switching events and a genuine loss-of-mains event. This is due to the variable trip time allowed, as the transient event is short and the phase change due to the load step or switching event is smaller than the trip threshold Φ_{Trip} = 20°.

- Applying a much smaller trip threshold ($\Phi_{Trip} < 7°$) would have risked a spurious trip. This emphasises the benefit of setting the trip threshold as high as reasonably practical and allowing the potential trip time to be as long as required by the local protection requirement or agreement with the PES. For switching events which cause nodal step phase changes, the trip setting must be larger than the maximum expected step phase change.

- A ROCOF relay would risk tripping during the same event, unless the qualification time is set to at least 100-200ms.

6.3 Application to loss-of-mains detection: sensitivity testing using real-time hardware/processing and different/novel DG control strategies

During the work leading to this thesis, a laboratory power systems network capable of performing experiments in microgrid control has been created by the author. A schematic is shown in Figure 6-28. The laboratory network contains a "Grid Supply Point" (GSP), which is the connection point to a "parent network". There are two options available for use as the parent network: the local 433V 3-phase mains supply can be used, or an 80kVA synchronous generator. The 80kVA synchronous generator is a more flexible device to use, since the frequency and voltage can be perturbed away from nominal values as desired (through sets of scenarios) to mimic problems within the parent network. The terminals of the 80kVA generator can also be synchronised with a real-time digital simulation (performed on an RTDS digital simulator, RTDS (2008)) of a much larger power network. Measurement of currents at the machine terminals can be fed back into the simulation (via simulated current sources) to provide a closed-loop simulation-hardware-simulation path. In this way, the remainder of the laboratory power network becomes "hardware-in-the-loop", with the 80kVA synchronous generator as the linking hardware.

For the loss-of-mains tests presented in this section, the national grid LV supply was used as the parent network. This is simply because it was quieter than using the 80kVA generator, and because the flexibility of the 80kVA generator as a parent network is not required for loss-of-mains testing.

The laboratory contains 2 separate microgrids containing distributed generation of both the synchronous and solid-state inverter style. Local loads can be configured on each microgrid to represent steady or fluctuating loads, both real and reactive. The microgrids are controlled by locally autonomous control algorithms, developed by the author and prototyped on the ADI RTS (Real-Time-Station) (ADI, 2008).

Energy electrical / Marine Systems
400V 3 phase AC Test Microgrid
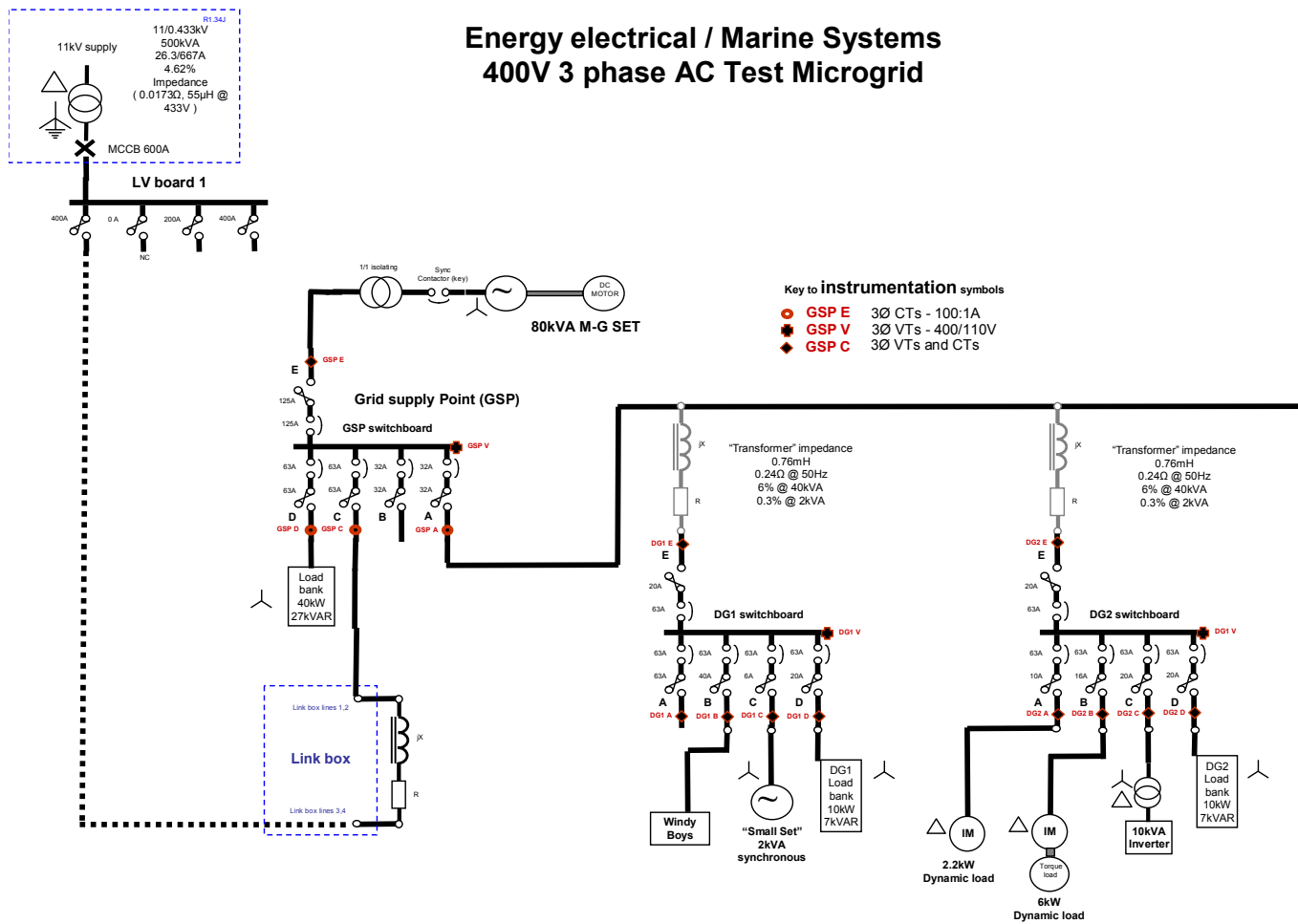
Figure 6-28 : 400V 3-phase microgrid laboratory; single-line diagram

293

6.3.1   Generator control strategies in grid-connected mode

Experience with the hardware of in Figure 6-28 confirms that it is relatively difficult to obtain a match between local generation set points and local load demands, in both P and Q, such that the LOM event cannot be detected within a timeframe of 2 seconds, when using the Phase Offset Relay (POR) of section 6.1. It is, however, possible to create such non-detection scenarios in the laboratory with synchronous generators:

- by deliberately creating an exact match between generator power output (both real and reactive) and local load demand within the network subjected to the loss-of-mains condition, AND

- if certain (un-recommended!) generator control strategies are used.

It has been found that the tripping time for DG-load systems which are well balanced varies depends heavily upon the exact control strategy used for the generator. The control strategy for the DG used by Dysko (2006) in section 6.1 is not known in detail. In the following sections, results from laboratory tests are shown for well-balanced and slightly off-balance cases, with different known control strategies used for the DG.

The four control strategies examined in detail are:-

1. Exact real & reactive power match between DG and local loads (no droops)

2. DG in PQ control with almost no droop (DG P&Q outputs essentially fixed)

3. DG in PQ control with 5% frequency and 10-40% voltage droop, which proves to be unstable in the accidentally islanded case. This instability greatly aids the detection of LOM, but does not completely eliminate the NDZ.

4. A novel microgrid control strategy to completely avoid the NDZ of LOM detection. This uses the DG in PQ control with no droop on the P control. The P control is used to obtain an exact active power match between DG output and local load demand. The Q target is set as desired, with a 10% voltage droop, plus an override algorithm which guarantees a non-zero VAR exchange with the parent network.

This section will show that strategies 1 & 2 can lead to sustained non-detection of a LOM event (and are hence not recommended!), strategy 3 leads to guaranteed detection of LOM but with potentially long detection times (up to 12 seconds in the laboratory), while strategy 4 can be used to guarantee LOM detection within a timeframe of 2 seconds.

It should be noted that control strategies for the DG which use drooped frequency and

voltage targets ("FV" type control) are not examined in detail here. It can be shown that such strategies can lead to very large LOM non-detection zones (sustained non-detection), since the controls are designed to be inherently stable in the islanded state. This type of DG control scheme is thus not appropriate when using local (communication-less) passive LOM detection relays.

Strategies 1 and 4 are of particular interest. There are times when deliberately obtaining a close match between DG output power and local load demand is desirable. The obvious example is during a planned change-over to an islanded state, which might be done to ensure security of supply or to improve power quality. Before changing to islanded mode, it is highly beneficial to pre-match DG real power output to the local load real power demand. When the actual change to islanded mode occurs, the frequency excursions are thus minimised as the prime mover throttle control is already at the correct set-point. A mismatch in real power leads to a sudden prime mover power output requirement upon islanding, and this may take several seconds to implement. This applies not only to rotating prime movers but also to static prime movers such as fuel cells etc. which may have inherent lag times due to fuel pumps and pressures. If the output power cannot be ramped quickly enough, the frequency will go outside allowable limits (or in the case of an inverter, the DC bus may collapse or over-volt). A similar pre-match in reactive power is also desirable, although the tolerance to mismatches is much higher as the effect upon islanding will be a brief voltage surge or depression. The tolerance for short-term voltage excursions is at least ±10%. Voltage can be also be adjusted relatively quickly (compared to throttle settings), via electronic field controls for synchronous generators or switching patterns for inverters.

Strategy 1 aims for a perfect local match of both real and reactive power, and thus provides no frequency or voltage support. The resulting microgrid is also deliberately placed within the likely non-detection zone of loss-of-mains relays. Strategy 4 is a novel scheme which deliberately matches real power locally but does not aim to match reactive power locally. Instead, it provides voltage support to the local and wider network by VAR exchange with the parent network, while ensuring a finite VAR exchange with the parent network to completely avoid the non-detection zone of the loss-of-mains relay.

6.3.1.1   DG control strategy 1 – Exact real & reactive power match between DG and local loads

Strategy 1 uses PQ control without any droop controls at all, but an active balancing algorithm. This simple strategy aims to source exactly the same amount of power from the DG as is required by the local load, both for real and reactive powers. This strategy was

implemented to automatically and deliberately create scenarios which might demonstrate the non-detection zone of loss-of-mains relays in the laboratory.

6.3.1.2   DG control strategy 2 – PQ control with almost no droop

Strategy 2 uses PQ control with 10000% frequency droop (for a 1pu change in real power output) and 10000% voltage droop (for a 1pu change in reactive power output). In this case, the DG outputs an almost constant amount of real and reactive power, independent of measured frequency and voltage. There is only a very small restorative effect towards nominal frequency and voltage, and the system is unstable when any significant perturbation arises.

6.3.1.3   DG control strategy 3 – PQ control with 5% frequency and 10-40% voltage droop

Strategy 3 uses PQ control with 5% frequency droop (for a 1pu change in real power output) and 10-40% voltage droop (for a 1pu change in reactive power output). In this case, the DG outputs real and reactive powers which tend to have a "restorative" effect on frequency and voltage towards nominal values via the droop controls. However, the droop controllers and generator/prime mover controls/response contain phase lags, which tend to push the system into an unstable state in islanded mode. Such a PQ control strategy has been used by the author in the laboratory at Strathclyde. It has proved to be an appropriate control strategy for a grid-connected DG unit.

To understand the instability of strategy 3 following a LOM event, the control strategy and system response can be approximately modelled. A simplified and approximate diagram of the control system and plant is shown below (Figure 6-29). The diagram is split into two parts: P control (throttle or real power) and Q control (field or reactive power). The controller is designed to operate in grid-connected mode controlling the export of real and reactive power with droop controls, with frequency and voltage set predominantly by the parent network. In the diagram below, the control system and plant is placed (accidentally) in islanded mode. Here, it has been assumed for simplicity that the active and reactive power requirement of the loads is fixed. The generator electrical output powers P_{Gen} and Q_{Gen} are therefore also fixed at the load real and reactive powers, since the generator and loads are joined together in an islanded power system. Also, the cross-couplings between the P and Q systems have been ignored in the stability analysis. Some of the additional linkages which would be required to model all the real effects are indicated by dotted lines and boxes. The following effects are thus ignored:-

- Load real powers proportional to voltage or voltage^2 (such as light bulbs), or frequency or frequency^2 (such as fans).

- Load powers dependent upon rates of change of frequency. This would occur due to loads with inertia, causing regeneration for example.

- Load reactive powers being dependent upon voltage, voltage^2, or frequency.

- The impact upon voltage of the generator frequency (due to the rotor field current creating more volts at the armature).
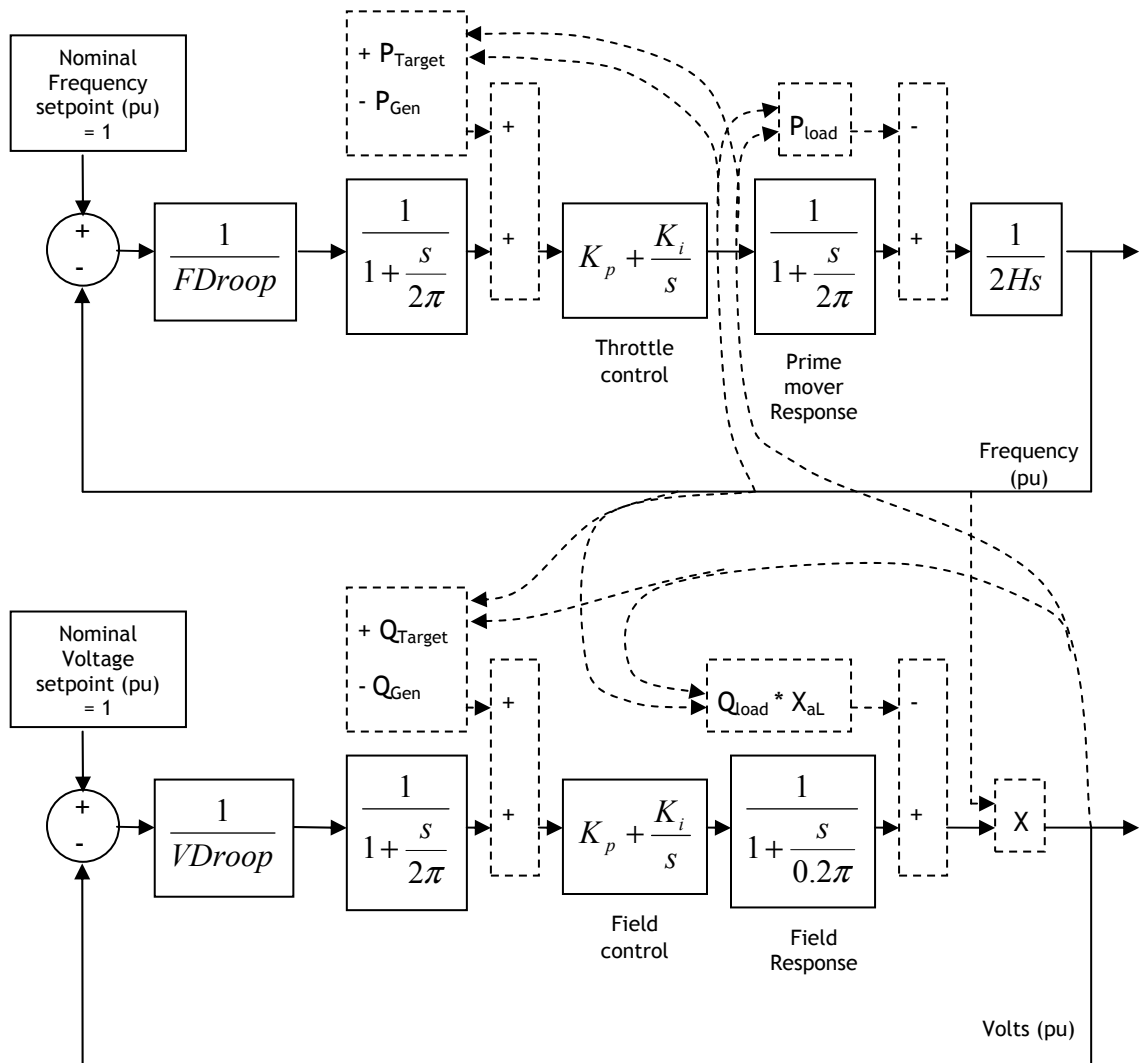
Figure 6-29 : Simplified diagram of control and plant during an unintentional island event. P-control and Q-control systems with droop.

A qualitative analysis of the simplified P control (real power) control loop shows that it consists of:-

- The inputs, which are the nominal frequency set-point (1pu) and the actual system frequency, against which the real power target is drooped.

- The drooped frequency setting *FDroop*, which is 0.05 (5%), giving a gain of 20.

- A low-pass filter implemented in software, set with a cut-off frequency of 1Hz. This smoothes noise and slugs the droop output. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies.

- The throttle control. Here, K_p is set to 0, and all the control is integral, with K_i=1. This has a gain of infinity at DC, $1/2\pi$ at 1Hz, and further decreasing gain with frequency. Phase lag is fixed at 90° for all non-DC signals. This phase lag is an important component of the OLTF which introduces instability during islanded operation.

- The prime mover torque/power response time, estimated here by a low-pass filter with a cut-off frequency of 1 Hz. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies. This represents a fast-responding prime mover and many large movers will have slower responses leading to higher instability in the following analysis.

- The prime mover (and hence system frequency) will speed up or slow down proportional to the power difference between primer mover power output and the load power. The rate of change is reduced by $1/2H$ where H is the prime mover / generator per-unit inertia. Here, H is estimated as 1, a suitable figure for a distributed generator (Mullane, 2005). This gives a gain of infinity at DC, $1/4\pi H$ at 1 Hz, and further decreasing gain with frequency.

The total P-control system open-loop transfer function thus has a gain of infinity at DC and $1/FDroop/16\pi^2 H$ at 1Hz, decreasing further with increasing frequency. This is a gain of 0.13 at 1Hz using the example values. The phase lag just above DC is 180°, rising to 270° at 1Hz, and increasing towards 360° at higher frequencies. This simple analysis is confirmed by the bode plot of the transfer function (Figure 6-30).

Referring to Figure 6-30, and using classical analysis of gain and phase margin, this system is unstable. The phase margin is non existent, because the phase lag of the OLTF (Open Loop Transfer Function) is equal to (and larger than) 180° over the range of frequency values from DC to 0.45Hz, where Gain is also >1. Similarly, there is no gain margin. This analysis predicts that the system will oscillate at some frequency below 0.45Hz. As will be seen in section 6.3.6, this is what is observed.
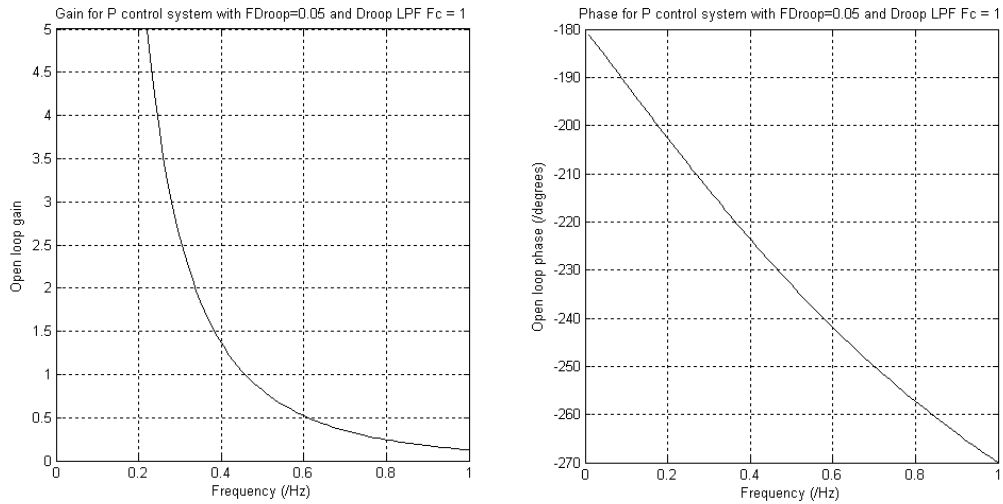
Figure 6-30 : Bode plot of simplified P (real power) grid-connected control loop during unintentional islanding, 5% frequency droop

A qualitative analysis of the simplified Q control (reactive power) control loop of Figure 6-29 shows that it consists of:-

- The inputs, which are the nominal voltage set-point (1pu) and the actual system voltage, against which the reactive power target is drooped.

- The drooped voltage setting *VDroop*, which is between 0.1 (10%) and 0.4 (40%), giving a gain of between 10 and 2.5 respectively.

- A low-pass filter implemented in software, set with a cut-off frequency of 1Hz. This smoothes noise and slugs the droop output. Gain is 1 at DC, and $1/\sqrt{2}$ at 1Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 1 Hz signal, and up to 90° for higher frequencies.

- The field control. Here, K_p is set to 0, and all the control is integral, with K_i=1. This has a gain of infinity at DC, $1/2\pi$ at 1 Hz, and further decreasing gain with frequency. Phase lag is a fixed 90° for all non-DC signals. This phase lag is an important component of the OLTF which introduces instability during islanded operation.

- The synchronous generator field current drive response time, estimated here by a low-pass filter with a cut-off frequency of 0.1 Hz (the approximate response of an 80kVA synchronous generator in the Strathclyde laboratory). Gain is 1 at DC, $1/\sqrt{2}$ at 0.1Hz, and about 1/10 at 1 Hz, decreasing further with increasing frequency. A phase lag of 45° will be added for a 0.1 Hz signal, and up to 90° for higher frequencies.

299

The total Q-control system open-loop transfer function thus has a gain of infinity at DC and approximately $1/VDroop/(10\sqrt{2}*2\pi)$ at 1 Hz, decreasing further with increasing frequency. This represents a gain of 0.028 at 1Hz for the 40% droop case. The phase lag just above DC is 90°, rising to about 225° at 1 Hz, and increasing towards 270° at higher frequencies. This simple analysis is confirmed by the bode plot of the transfer function for the 40% droop slope case.
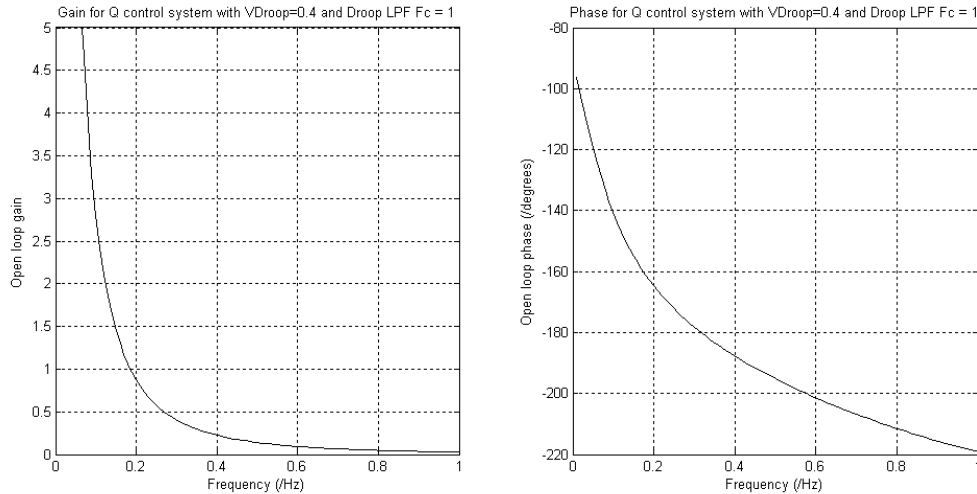
Figure 6-31 : Bode plot of simplified Q (real power) grid-connected control loop during unintentional islanding, 40% voltage droop

Referring to Figure 6-31, and using classical analysis of gain and phase margin, this system may just be stable. However the phase margin is only 20° and the gain margin is 9dB. The approximations made during the analysis are large enough that in reality the system could easily be unstable in isolation. This is especially true when the cross-couplings to the unstable P-control system are considered.

Changing the voltage droop from 40% to 10% results in a 4x (12dB) increase in the OLTF gain. The bode plot for the resulting system is shown in Figure 6-32. This system is now unstable in isolation, as was the P-control system. The phase margin is non existent, because the phase lag of the OLTF (Open Loop Transfer Function) is equal to (and larger than) 180° over the range of frequency values from 0.31Hz to 0.38Hz, where Gain is also >1. Similarly, there is no (-3dB) gain margin. This analysis predicts that the system will oscillate at some frequency between 0.31 and 0.38Hz.
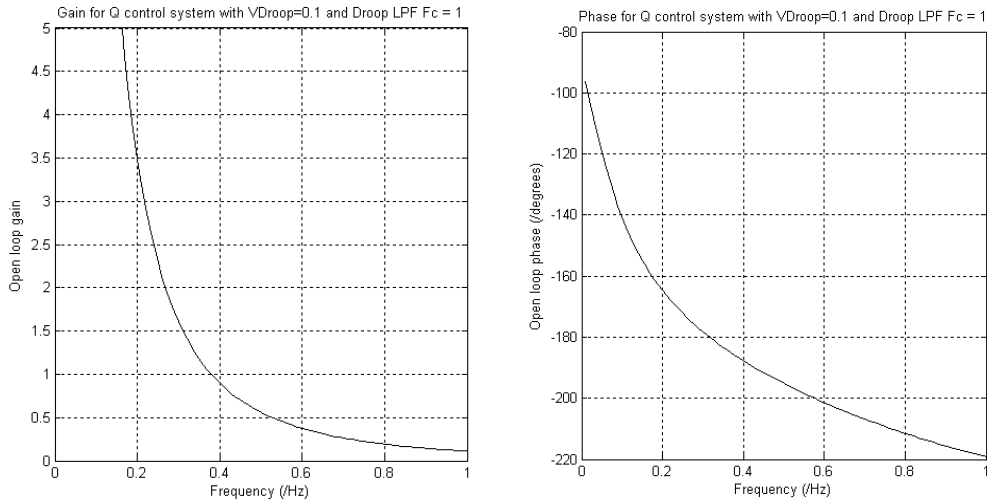
Figure 6-32 : Bode plot of simplified Q (real power) grid-connected control loop during unintentional islanding, 10% voltage droop

Works from several authors such as Du (2005) and Ye (2006) also advocate the use of grid-connected controls for DG which become fundamentally unstable when the power system becomes islanded. In these works, the controllers are adapted from frequency and voltage (FV) controllers (with P and Q offsets) using proportional or PI controls which are inherently stable in islanded mode. These can only be made unstable by adding positive feedback terms via bandpass filters. The advantage of strategy 3 (and 4) over such previously published works is that in strategy 3 & 4 the traditional droop control settings are retained intact, so that network support and power/VAR sharing functions are maintained, without the addition of extra feedback terms. This allows the DG unit to partake in frequency and voltage support, via modified P and Q exports dependent upon measured system frequency and voltage. The instability arises purely from the natural instability of this system (when unintentionally islanded) due to phase lags (particularly due to the integral-only controls) and droop slopes (loop gains).

6.3.1.4 DG control strategy 4 – Real power match and 10% voltage droop, guaranteed LOM NDZ avoidance using non-zero VAR exchange

Strategy 4 uses PQ control based on strategy 3 but with four significant modifications:-

- Generator real power output can be deliberately matched to the local load demand, by an automatic process without a droop control.

- The reactive power export target from the DG is set to 0.

- Voltage droop is set at 10%, to provide significant voltage support, and also to guarantee that the Q control system is independently unstable (by the analysis of section 6.3.1.3 and Figure 6-32).

- A simple over-ride algorithm always insists on a measurable VAR exchange (more than about ±0.05pu) with the parent network. This guarantees that exact PQ balance within the local microgrid is never obtained and the loss-of-mains event is detectable within a reasonable timeframe (Affonso, 2005), even when an exact active-power balance is achieved.

This method actively makes a balance of real power generation and demand, driving the real power exchange with the parent network to zero. The reactive power import/export will adjust itself, via the 10% droop control, in order to provide a stabilising effect on voltage towards the nominal voltage level (it provides significant voltage support, up to 1pu VAR export/import if the voltage varies from nominal by 10%). A zero reactive power exchange with the parent network is avoided via a strategy that adjusts the reactive power flow, and is simple in concept. Care has to be taken to add appropriate hysteresis within the algorithm, however, to stop it cycling around decision thresholds. In some respects the algorithm is similar to that proposed by Lokov (2005), however in that work the active power exchange with the parent network is deliberately moved from zero, while the reactive power is not considered. The algorithm proposed here adjusts reactive power flow away from zero, and hence allows much more freedom of generator set-point selection, particularly with respect to active power.

This novel control strategy involves an algorithm with several steps. This algorithm can be executed continuously in real time while a generator (and its local power system including loads) is grid-connected to a parent network. The algorithm is the subject of UK patent application 0810512.4 filed 10 June 2008 (Roscoe, 2008).

A diagram of the context is shown in Figure 6-33.

The algorithm ensures that either real power or reactive power exchange (P_{Net} and Q_{Net}) between the local power system and the parent network are above certain thresholds. These thresholds are small in per-unit terms. However, they are large enough that, upon unintentional islanding, the local power system control is always perturbed enough that the unstable controls of generator real and reactive power outputs result in oscillation and detectable loss-of-mains conditions within less than 2 seconds (Affonso, 2005).
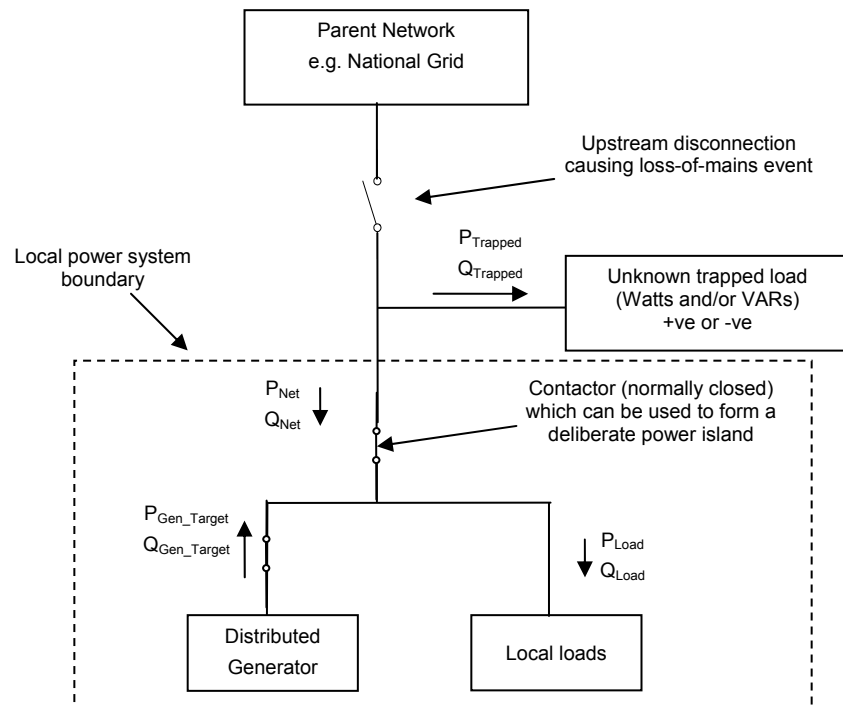
Figure 6-33 : **Electrical single-line diagram of the context for control strategy 4**

In simple terms, the algorithm is described below:-

1)   Either: intentionally match the local active power generation target P_{Gen_Target} with local real power demand P_{Load}, or, determine the target P_{Gen_Target} appropriately using financial, power flow or operational drivers/constraints. This optional power matching can be an important precursor to deliberately islanded operation, because it minimises the frequency and prime mover transients when a change from grid-connected to islanded operation occurs. However, matching the active power target also means that the local generation tends to track any local load changes and this may result in prime mover wear or non-optimal efficiency due to the constant adjustment of throttle and power output levels. Some of this effect can be mitigated by suitable dead-band functions within the prime mover throttle control. The decision on whether to match the active power target must be taken by appropriate trade-offs between the risks of outage versus the costs of operation in the different modes. These inputs may change dynamically in real time, requiring a constant re-assessment of the optimal operational mode.

2)   Monitor the real power exchange from the parent network to the local power system. Call this power flow P_{Net}.

3)   If the absolute value of P_{Net} falls below a set threshold P_t, then there is a close

303

match of real power between local load demand and local generation output (excluding trapped load), and the flag P_{Match} is set to TRUE (otherwise P_{Match} is set to FALSE).

4) If P_{Match} is FALSE, then the algorithm is finished and starts again at step 1 in the next execution frame.

5) Otherwise, if P_{Match} is TRUE, continue …

6) Monitor the reactive power flow from the parent network to the local power system. Call this power flow Q_{Net}. In practice, due to hysteresis considerations within the following control algorithm, Q_{Net} should not be measured directly, but instead deduced by subtracting the normal generator VAR output target Q_{Gen_Target} (after the set point Q target and droop controls have been applied) from the measured local load reactive power demand Q_{Load}.

7) If the absolute value of Q_{Net} falls below a threshold Q_t, then there is (or will be) a close match of real and reactive power between local load demand and local generation output (excluding trapped load). The flag Q_{Adjust} is set to TRUE.

8) If Q_{Adjust} is FALSE, then the algorithm is finished and starts again at step 1 in the next execution frame. If Q_{Adjust} is TRUE, then continue ….

9) The aim is now to adjust Q_{Gen_Target}, the target reactive power output from the generator, such that Q_{Net}, the reactive power flow from the parent network to the local power system, has an absolute value of plus or minus Q_t, the threshold below which we do not want abs(Q_{Net}) to fall. Decide which way to adjust Q_{Gen_Target} (up or down) in order to achieve abs(Q_{Net})$> Q_t$ with the minimum of adjustment to the original generator reactive power target Q_{Gen_Target}. The decision can be made by using a flag Q_{Adjust_Up}=(Q_{Gen_Target}-Q_{Load}>0). This means that if Q_{Gen_Target}>Q_{Load}, Q_{Adjust_Up} will be TRUE, or FALSE otherwise. The idea here is that if Q_{Gen_Target}>Q_{Load}, reactive power is already flowing from the local power system back into the parent network (but the magnitude of the reactive power flow is less than Q_t as already determined). Thus, in this case, increasing the reactive power output power from the generator by less than Q_t will cause abs(Q_{Net}) to exceed Q_t which is the desired result. If the generator output was reduced, then it would have to be reduced by more than Q_t to achieve abs(Q_{Net})>Q_t.

10) The modified generator reactive power output target can now be calculated from $Q_{Gen\_Target\_New} = Q_{Load} + Q_t$ (if Q_{Adjust_Up} is TRUE), or $Q_{Gen\_Target\_New} = Q_{Load} - Q_t$ (if Q_{Adjust_Up} is FALSE)

11) A final check is that $Q_{Gen\_Target\_New}$ is within the acceptable control range of the generator. If not, then the setting of Q_{Adjust_Up} should be inverted and the value of $Q_{Gen\_Target\_New}$ recalculated.

An overview of this process (not including all the details) is summarised graphically in Figure 6-35. This process is repeated at a suitable frame rate within a microgrid controller.
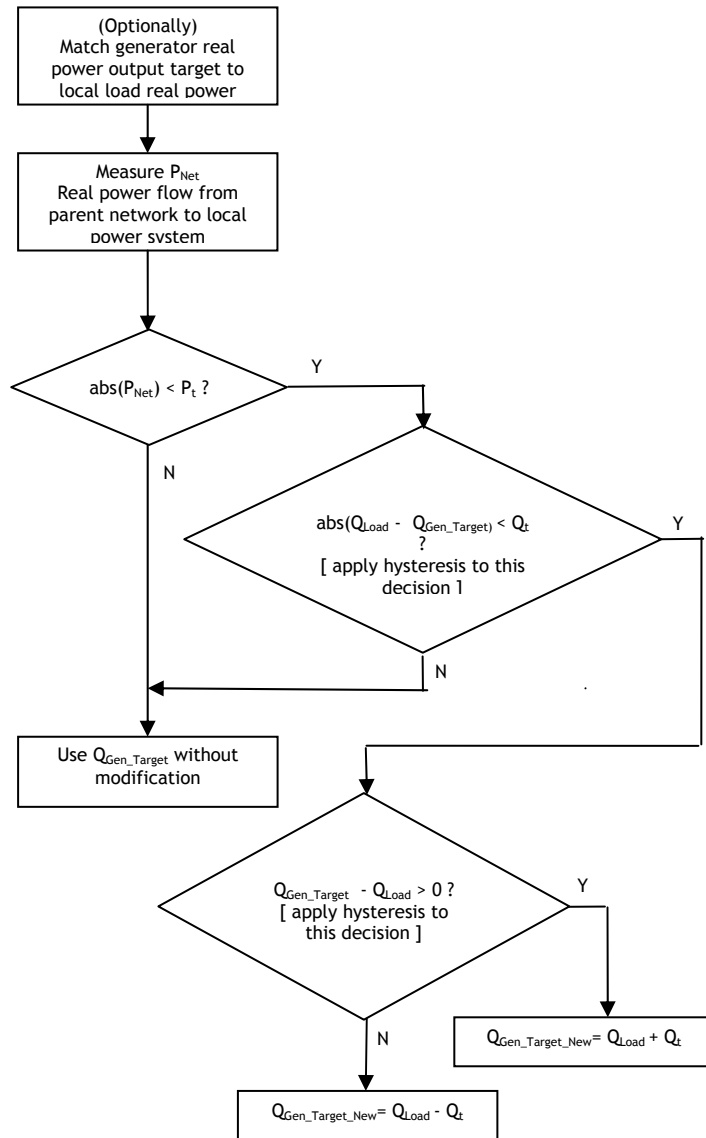
Figure 6-34 : Abbreviated summary of strategy 4 algorithm

It should be noted that in addition to the steps described above, appropriate hysteresis (involving time and/or decision thresholds) should be applied during the decision-making processes which set the Boolean values of

- P_{Match} (Boolean decision)
- Q_{Adjust} (Boolean decision)
- Q_{Adjust_Up} (Boolean decision)

This hysteresis avoids the controls changing regularly from 0 to 1 and vice versa when the

active and reactive power flows are hovering around the decision thresholds. There is also some value in converting the Boolean value Q_{Adjust_Up} (with value 0 or 1) into a floating point value $Q_{Adjust\_Direction}$ with a value of -1 or +1 respectively. This can then be passed through a slew-rate filter with appropriate slew rate limits to give $Q_{Adjust\_Direction\_Rate\_Limited}$. This slew-rate limiting simply smoothes out any step changes to generator reactive power targets which would otherwise occur. $Q_{Gen\_Target\_New}$ can then be calculated from

$$Q_{Gen\_Target\_New} = Q_{Load} + Q_t{}^* Q_{Adjust\_Direction\_Rate\_Limited}$$

A final comment is that there may be trapped loads outside the boundary of the local power system, as shown on Figure 6-33. If a loss-of-mains event occurs, it may result in load (or generation) outside of the local power system becoming part of the unintentional power island.

The worst case would then be if the local active power P_{Gen} is not deliberately matched to P_{Load} and is in fact accidentally almost matched to (P_{Load} + $P_{Trapped}$), and also if a close reactive power match accidentally exists between Q_{Gen} and (Q_{Load} + $Q_{Trapped}$). This is an unlikely but potential scenario. Note that this scenario can be avoided by deliberately matching P_{Gen} to P_{Load}, as in the optional step 1) above. This means that if $P_{Trapped}$ is significantly non-zero, i.e. abs($P_{Trapped}{>}P_t$), then P_{Gen} will never be approximately equal to (P_{Load} + $P_{Trapped}$) since P_{Gen} = P_{Load}. If $P_{Trapped}$ is very close to zero, i.e. abs($P_{Trapped}$)$\leq P_t$, then P_{Gen} will be very close to (P_{Load} + $P_{Trapped}$) but in this case the algorithmic steps 2) to 11) above will take place. For an accidental close match of reactive power $Q_{Gen} \approx$ (Q_{Load} + $Q_{Trapped}$) to then also occur, abs($Q_{Trapped}$) would then have to be $\geq Q_t$, . This is unlikely if abs($P_{Trapped}$)$\leq P_t$, unless the trapped load (or generator) has an extremely poor power factor. Thus is can be seen that, (counter-intuitively), operating the local power system with a deliberate match of local real power generation to local power demand can be used as a tool to avoid the non-detection zone of loss-of-mains, when the possibility of additional trapped loads exists.

Referring back to the system model for strategy 3 of section 6.3.1.3, in Figure 6-29, the control scheme for strategy 4 can be compared to it.

- the Q control scheme is identical except that the droop is changed from 40% to 10%, and the system is initially deliberately destabilised by the guaranteed VAR exchange.
- The P control scheme is different, due to the active power matching algorithm.

The first pass simplified control model for strategy 4 can be represented as shown below. This is adapted from Figure 6-29.
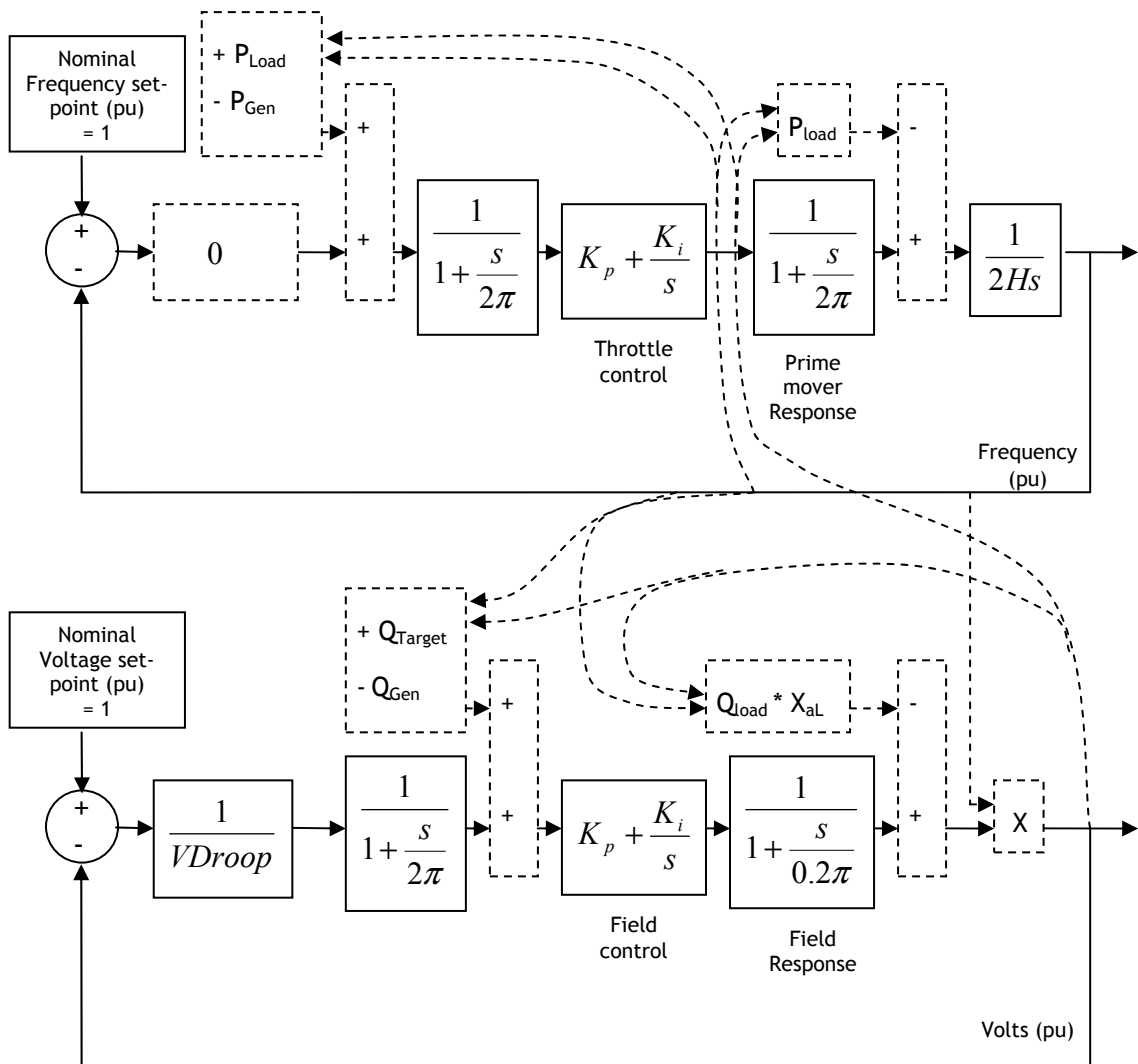
Figure 6-35 : Simplified diagram of control and plant during an unintentional island event. Active power match and drooped Q-control systems.

The bode plot and stability analysis for the Q droop system is identical to that shown in Figure 6-32. It is unstable and is likely to oscillate at 0.31 to 0.38Hz.

The P control loop, on the other hand, has been broken. The loop gain is now zero. Instead, the input to the PI controller for throttle is the difference between the measured load power and the measured generator output power. During normal grid-connected operation, the load power can be measured by adding the generator output power to the power fed by the parent network. Thus, the input to the PI controller is given by:-

$$PI\_Controller\_Input = P_{Load} - P_{Gen} = (P_{Gen} + P_{Net}) - P_{Gen} = P_{Net}$$

<div align="right">(6.5)</div>

Therefore, during grid-connected operation, P_{Net}, the power flow from the parent network plus the trapped load adjustment, will be driven to zero by control action. However, when a loss-of-mains event occurs, P_{Net} becomes $-P_{Trapped}$, defined by the actual real power demand of any trapped load. Thus, the PI controller input will be $-P_{Trapped}$ by (6.5).

Similarly to control method 1, of section 6.3.1.1, the P control loop when analysed alone is not classically unstable when accidentally islanded due to LOM event. However, the loop gain is zero and thus there is no control at all. Any destabilising of the system (for example a small local load step, or $P_{Trapped} \neq 0$) will not be corrected by any restorative action, the input to the PI controller will be a constant non-zero number and power output will rise or fall in an approximately linear slope due to integral control action. When combined with an unstable Q control and the P-Q control cross-couplings, overall instability will almost certainly result. This causes frequency deviations allowing detection of the LOM event, and minimisation of the NDZ.

6.3.2 Laboratory testing of the POR with the 4 different DG control strategies

To assess the performance of both the POR and the different proposed DG control strategies, the laboratory microgrid of Figure 6-28 was used. The amplitude, phase, frequency, and POR algorithms developed in this thesis were integrated into a much larger set of algorithms which are designed to control an entire microgrid containing a single major despatchable DG unit and local loads (loads which may contain smaller despatchable or renewable DG units effectively contributing negative loads). This microgrid control application has been developed by the author, but to describe it in full is beyond the scope of this thesis. One of the major challenges within this application is the successful detection of loss-of-mains (LOM) within a microgrid scenario, particularly when the microgrid DG and local load powers are deliberately matched during a pre-islanding process as described in section 6.3.1. Since the microgrid control algorithm is designed to operate with either synchronous or inverter-connected generation, the LOM-detection algorithm is desired to be a passive method rather than an active method.

The microgrid control algorithm, and other related code to operate the entire network, is written in MATLAB Simulink (with some S-functions), auto-generated into "C" code, and then built into executables which run on a Real-Time-Station (RTS) (ADI, 2008). The main microgrid control application runs on a single CPU at a 500 Sa/s frame rate (nominally 10

samples per cycle). Other lab infrastructure, data logging etc. runs on parallel processors at the same frame rate. IO (Inputs/Outputs) is handled on a separate processor. This can be clocked at 1500Sa/s (30 samples per cycle, 3x oversampling) and FIR filters applied in a similar method to that described in section 4.5.

In the sections which follow, for each proposed DG control strategy, the following process (with small variations) was undertaken:

- Connect the microgrid loads to the national grid. The grid connection is directly via a 500kVA 11kV/433V transformer. The microgrid loads in the synchronous generator case were approximately 1300-1400W at a power factor of 0.9 lagging. This power level was limited by the synchronous generator rating. In the inverter case the loads were larger, due to the inverter having a larger capacity of 10kVA.

- Synchronise the generator to the microgrid (and the national grid). This a reliable automated process, built into the microgrid control application.

- Set droop controls as appropriate for the control strategy under test.

- Set P and Q set-points to achieve a very close balance in both real and reactive power, such that the P and Q exchange with the parent network (the national grid) become very small. Note that for methods 1 and 4 (see section 6.3.1) this balancing method is automatic.

- Begin data logging

- Instigate a deliberate loss-of-mains condition by opening a contactor ("GSP A" on in Figure 6-28) upstream of the microgrid

- Wait for LOM to be detected …

- Stop data logging

- Repeat …

The data logging captures all variables at a 250Sa/s frame rate (decimation 2 from the 500Sa/s main algorithm frame rate). Nodal voltages and currents can be captured at the full 500Sa/s frame rate if necessary, and subsequently re-analysed with new candidate measurement algorithms.

In all cases, the trip setting of the POR was 20°. The ROCOF trigger threshold was set to 0.2Hz/s for most tests, except for some of the tests of strategy 4 in section 6.3.7. The inertia of the synchronous motor-generator is approximately H=0.9 pu.

6.3.3 DG control strategy 1 – Exact real & reactive power match between DG and local loads.

As described in section 6.3.1.1, control strategy 1 deliberately aims for an exact balance between DG real & reactive power outputs and the local load demands. As such it deliberately places the local microgrid in an extremely vulnerable state as regards the potential for non-detection of the loss-of-mains condition. The likelihood of being within the non-detection zone and either having a no-trip or very long trip result is high.

In the laboratory, it was indeed shown that a sustainable power island was able to be formed, without detectability of loss-of-mains[1]. The experiment was repeatable. This clearly shows that control strategy 1 is not inherently unstable (although any change to local load demand will cause it to be). Strategy 1 is thus not recommended for DG installations where LOM must be detected using passive relays based upon voltage measurements.

Figure 6-36 to Figure 6-41 show the results of the test which shows sustained non-detection of LOM. At the start (t=520), the DG and local loads are grid-connected, but the DG is outputting 1000W in a drooped manner. At t≈525 seconds, a "PQ" balance algorithm was engaged. The DG real and reactive powers then rise to meet the load powers (plus trapped load). The LOM event itself was instigated at t≈534 seconds, and is not detected. After this, the undetected power island frequency and voltage stay almost constant for many seconds. Only after a deliberate 150W load change is made at t≈594.75 seconds is the LOM detected (at t≈595.3 s, a 550ms trip time).
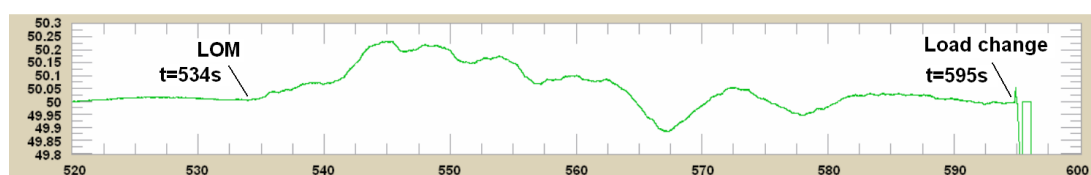
Figure 6-36 : Sustained non-detection of LOM; Frequency (Hz)

---

[1] To achieve this result, it was necessary to account for 20W of parasitic trapped load upstream of the microgrid, which was still back-fed during the LOM condition. The 20W offset had to be artificially inserted into the control loop. This parasitic load was measured by experiment, and consists of neon power indicator lamps and voltage transformer loads. The same 20W control offset was subsequently included in all the tests of this section, to deliberately create the biggest risk of non-detection.
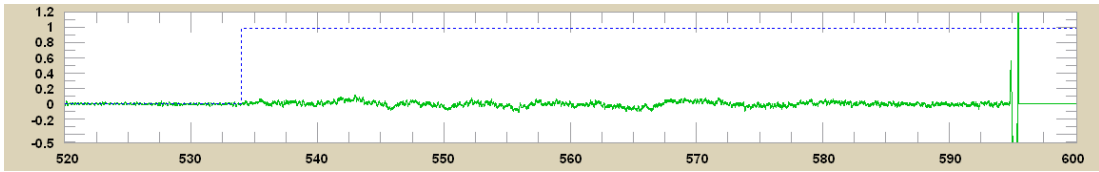
Figure 6-37 : Sustained non-detection of LOM; ROCOF (Hz/s) and instant of LOM inception (blue)
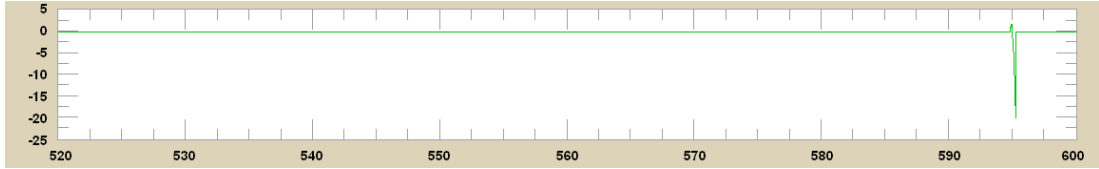
Figure 6-38 : Sustained non-detection of LOM; Phase Offset (degrees)
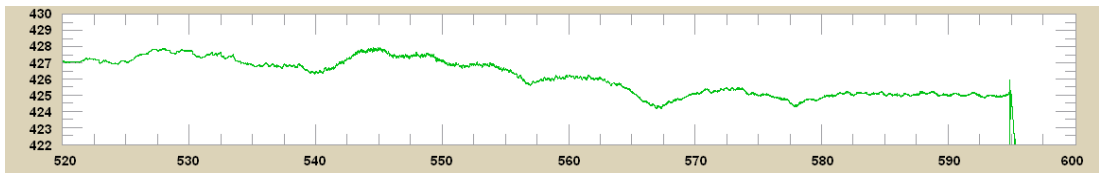
Figure 6-39 : Sustained non-detection of LOM; Voltage (+ve sequence fundamental, line-line)
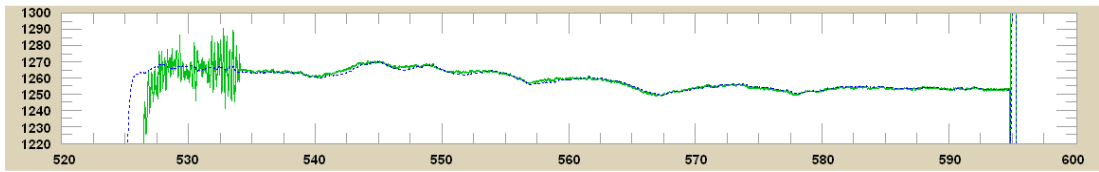
Figure 6-40 : Sustained non-detection of LOM; DG Real power output (solid) and target (dashed)
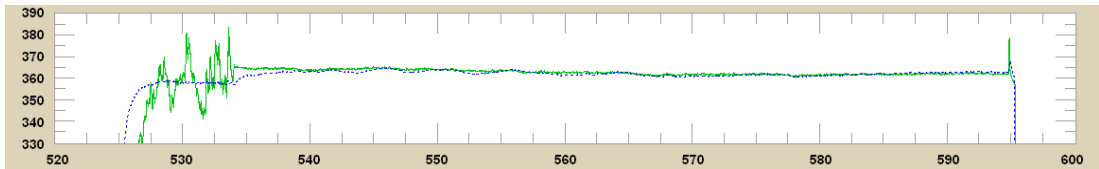
Figure 6-41 : Sustained non-detection of LOM; DG Reactive power output (solid) and target (dashed)

6.3.4 DG control strategy 1 – Exact real & reactive power match between DG and local loads, using inverter-connected generation

The above test method was repeated, but instead using a 10kVA inverter-connected generator and a suitable load magnitude. Despite the careful matching of both real and reactive powers[1], it was not possible to create a non-detection of loss-of-mains. This is due to the dynamics of the inverter control in grid-connected mode, and in particular the high bandwidth of the PLL which determines its effective inertia. This is extremely small, and hence the frequency excursion is much larger than for a synchronous generator when a LOM event occurs. The longest detection time captured was 1.4 seconds. This is shown below. The LOM event was instigated at t = 366.18 s and detected at t = 367.58 s.

This shows that (in this case), the use of inverter-connected generation can reduce the size of the LOM NDZ, due to the reduced "inertia" of the inverter hardware. This can be used to advantage in such installations. However, the inertia of a different inverter-connected system might be larger, either to different inverter software leading to a lower-bandwidth PLL (higher effective inertia), or due to high-inertia loads connected. Either of these two effects would significantly alter the results from those shown below, and potentially lead to sustained non-detection as shown in section 6.3.3.
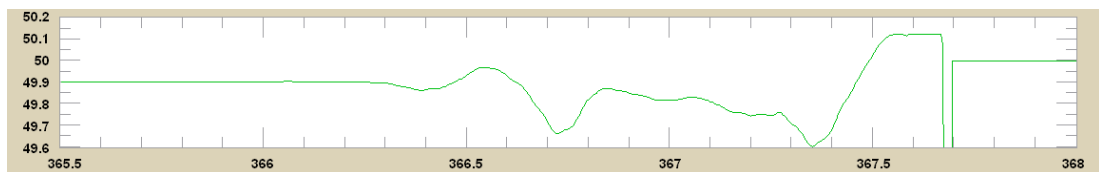
Figure 6-42 : Longest detection time using inverter-connected generation; Frequency (Hz)
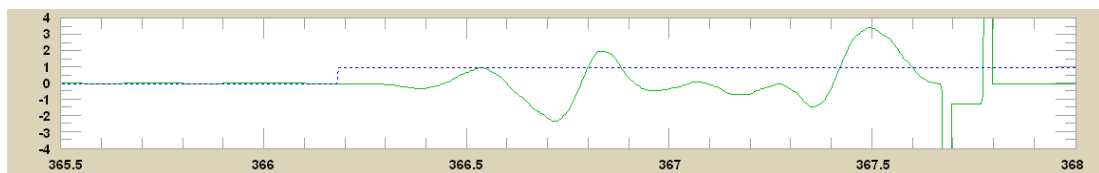
Figure 6-43 : detection time using inverter-connected generation; ROCOF (Hz/s) and instant of LOM inception (blue)

---

[1] A small amount of measured parasitic trapped load (8W in this case) was also accounted for
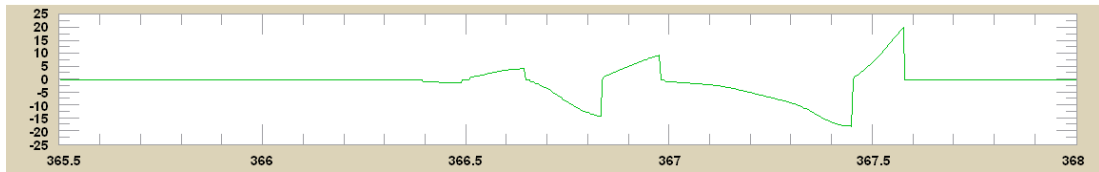
Figure 6-44 : detection time using inverter-connected generation; Phase Offset (degrees)
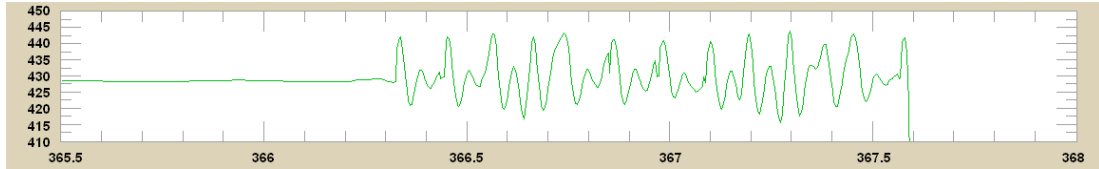
Figure 6-45 : detection time using inverter-connected generation; Voltage (+ve sequence fundamental, line-line)
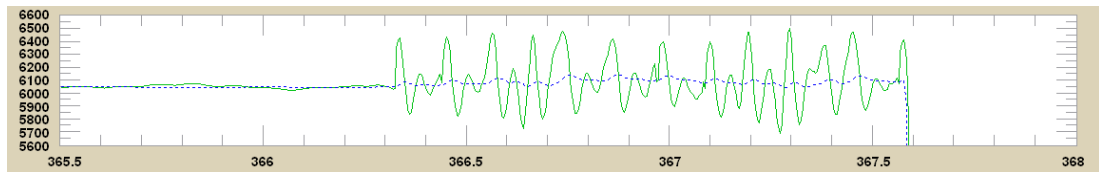
Figure 6-46 : detection time using inverter-connected generation; DG Real power output (solid) and target (dashed)
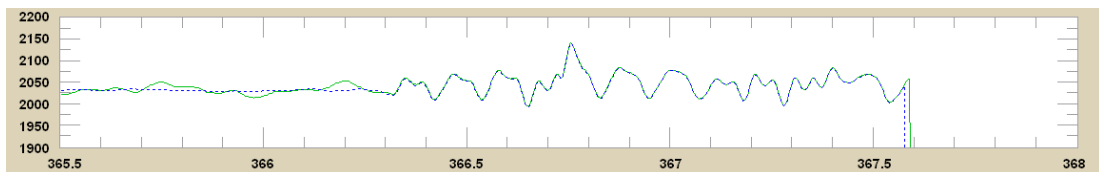
Figure 6-47 : detection time using inverter-connected generation; DG Reactive power output (solid) and target (dashed)

6.3.5   DG control strategy 2 – PQ control with almost no droop

These tests use the DG control strategy 2 of section 6.3.1.2. The P and Q set-point of the DG unit was manually set such that the real and reactive power exchange with the parent network was as close to zero as possible[1]. Since the target is a zero (or almost zero) power flow in the grid-connection branch, tuning the setup to achieve this is not heavily dependent upon exact calibration of any CTs, VTs, or sampling hardware. This is very helpful in achieving the balance accurately.

Using this control strategy, a case of sustained non-detection of LOM was demonstrated in the lab after only a few tries. The condition was sustained for >100 seconds until a small

---

[1] Actually, that the real power exchange was 20W to account for the expected trapped load, measured in section 6.3.3

313

load change was deliberately made. This caused very fast detection of LOM.

Strategy 2 is thus not recommended for DG installations where LOM must be detected using passive relays based upon voltage measurements. There are neither frequency/phase nor voltage excursions significant enough to enable the detection of LOM.

The graphs Figure 6-48 to Figure 6-53 below show logged data from the experiment. The loss-of-mains (LOM) event is instigated at t=236.5 seconds. Real power was matched within about 10W (<0.01pu) and reactive power was matched within about 10VAR (<0.01pu). At t=240 s, a LOM detection almost occurs, but the phase offset does not reach the trip level of 20° before the POR triggers are reset by a reversal of ROCOF. The undetected island, with the generator running at fixed P and Q outputs, was obviously not exactly balanced, because it settles to a new frequency of 49.2 Hz (down from 50.05 Hz) and voltage of 436V (down from 438V). The resulting state appeared to be perpetually stable in the laboratory, provided no changes to generator or load settings were made. A 150W load change was made at t=366.9 s. The POR tripped at t=367.4 s, 500ms after the small load change.
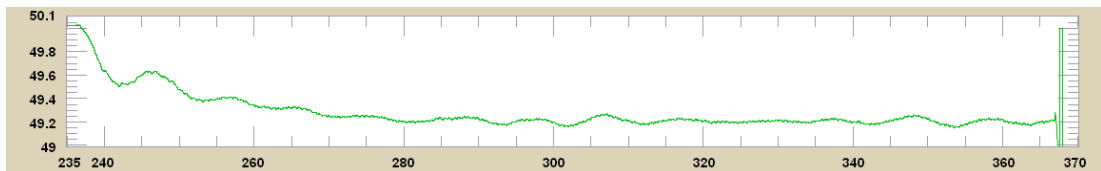
Figure 6-48 : Sustained non-detection of LOM; Frequency (Hz)
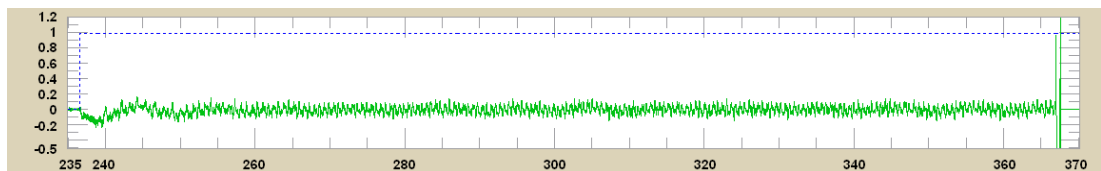
Figure 6-49 : Sustained non-detection of LOM; ROCOF (Hz/s) and instant of LOM inception (blue)
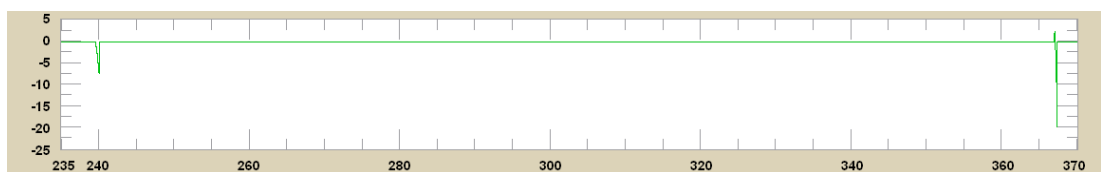
Figure 6-50 : Sustained non-detection of LOM; Phase Offset (degrees)
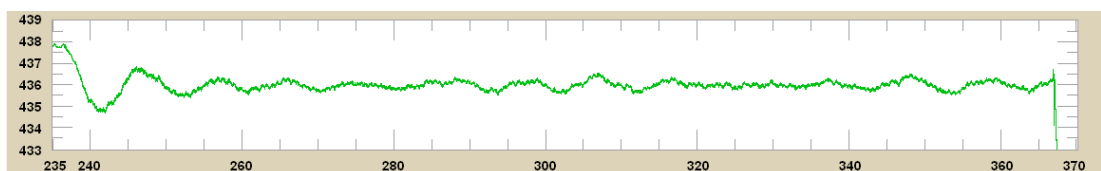
Figure 6-51 : Sustained non-detection of LOM; Voltage (+ve sequence fundamental, line-line)

314
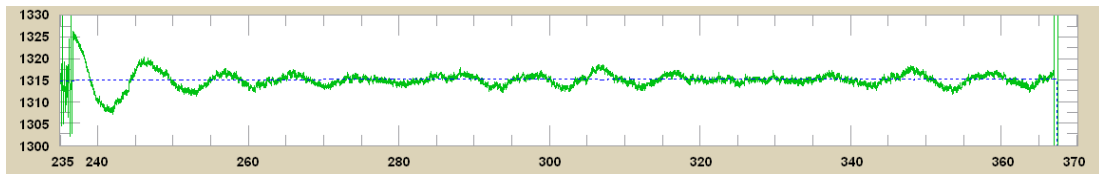
Figure 6-52 : Sustained non-detection of LOM; DG Real power output (solid) and target (dashed)
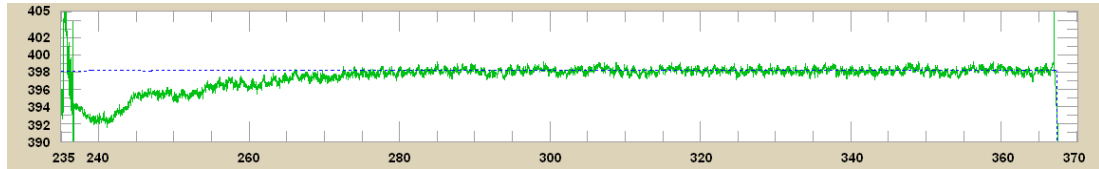
Figure 6-53 : Sustained non-detection of LOM; DG Reactive power output (solid) and target (dashed)

6.3.6   DG control strategy 3 – PQ control with 5% frequency and 40% voltage droop

These tests use the DG control strategy 3 of section 6.3.1.3. There is a 5% frequency droop and 40% voltage droop. The P and Q set-point of the DG unit was manually set such that the real and reactive power exchange with the parent network was as close to zero as possible[1]. This was quite hard to achieve in practice, because the parent network frequency and voltage were constantly changing, resulting in fluctuations to the generator output power (real and reactive) due to the action of the droop controls. During the tests performed, frequency often changed by several hundredths of a Hz during a few seconds. With the 5% frequency droop slope and a 1500W base power, a change of 0.03 Hz over 10 seconds results in a power adjustment of 0.012pu, or 18W. The voltage drifted by up to 1V (in 433) over 10 seconds, giving rise to reactive power adjustments of about 0.006 pu, or 9 VARs. Getting matches closer than this was difficult, but a number of attempts were made to get as perfect a match as possible.

21 attempts were made at achieving a perfect match. No sustained non-detections of LOM were noted. However, the longest trip time with the droop settings of 5% and 40% was 11.5 seconds although it will be seen in below that this event was a slight corner case, involving clipping of the generator output power which stopped the droop controls acting properly. The next highest trip times were 7.5 seconds and 5.5 seconds. The average trip time of the 20 trials was 3.7 seconds. The spread of trip times is shown in Figure 6-54.

---

[1] Accounting for the measured 20W parasitic trapped load
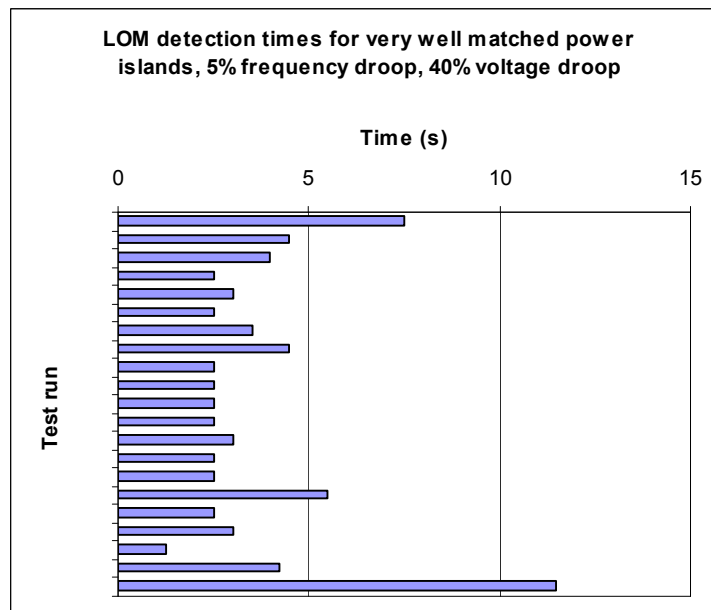
Figure 6-54 : LOM detection times for very well matched power islands, 5% frequency droop, 40% voltage droop

Strategy 3 is thus a significantly better control method than strategy 1 or 2, for DG installations where LOM must be detected using passive relays based upon voltage measurements. The LOM event will always be detected, but the detection time will be up to 12 seconds as suggested by laboratory testing. Although unlikely, the detection time might even be more than 12 seconds for extremely well balanced events, but only if the loads and DG set-points are extremely invariant.

An example of a slow detection with a trip time of 5.5 seconds is shown below. Real power was almost perfectly matched. Reactive power was matched to within 15 VARs (<0.01 pu). The LOM condition was instigated at t=919.7 s. Trip did not occur until t=925.2 s. The natural frequency of the system instability is approximately 0.31 Hz, shown by a half-period of about 1.6 seconds evident in Figure 6-59. This shows fair corroboration with the prediction made in section 6.3.1.3, which predicted instability with a natural frequency of between DC and 0.45Hz, based upon the P control system being unstable and the Q control system being marginally stable with a voltage droop of 40%. This provides evidence that the models produced in section 6.3.1.3 provide a fair means of analysing the system.
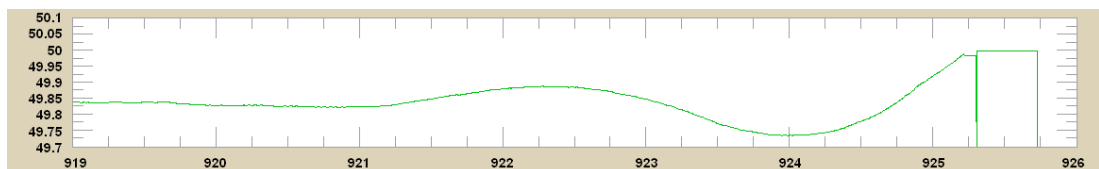
Figure 6-55 : 5.5s trip time with 5% frequency and 40% voltage droop; Frequency (Hz)
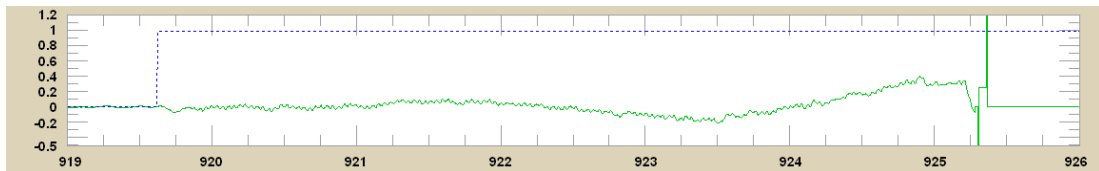
Figure 6-56 : 5.5s trip time with 5% frequency and 40% voltage droop; ROCOF (Hz/s) and instant of LOM inception (blue)
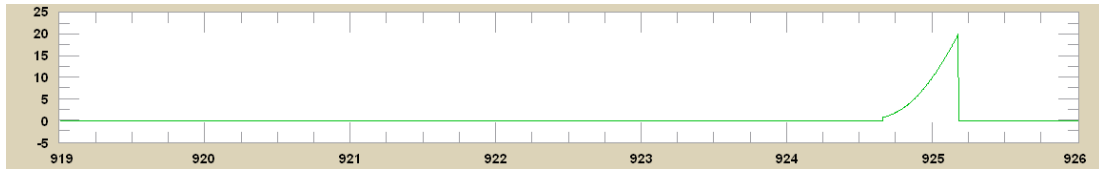
Figure 6-57 : 5.5s trip time with 5% frequency and 40% voltage droop; Phase Offset (degrees)
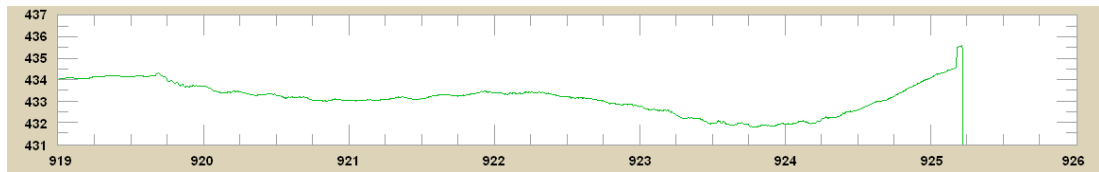
Figure 6-58 : 5.5s trip time with 5% frequency and 40% voltage droop; Voltage (+ve sequence fundamental, line-line)
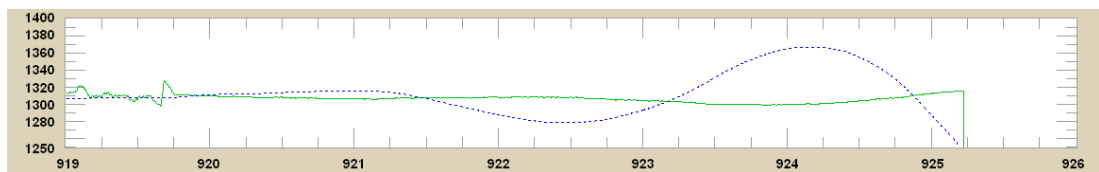
Figure 6-59 : 5.5s trip time with 5% frequency and 40% voltage droop; DG Real power output (solid) and target (dashed)
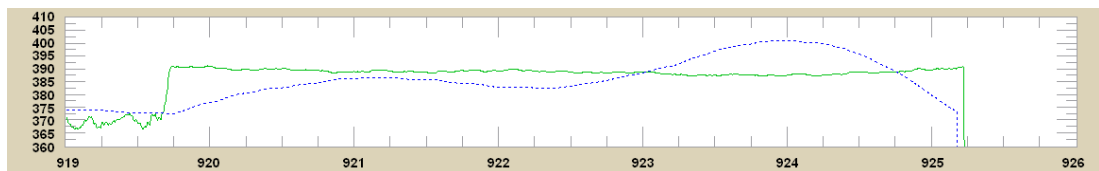
Figure 6-60 : 5.5s trip time with 5% frequency and 40% voltage droop; DG Reactive power output (solid) and target (dashed)

The example with a trip time of 11.4 seconds is shown below. Real power was almost perfectly balanced, while reactive power was balanced within about 10 VARs (<0.01pu). The LOM condition was instigated at t=743.3 s. Trip did not occur until t=754.7 s. This example is an interesting corner case. In this test, the local load power was extremely close to the rated output power of the generator (1500W). Figure 6-65 shows that the generator real power control (throttle) was clipped to 1500W (1pu). This diminished the action of the droop control feedback loop and consequently, the instability of the system was less and the trip time took longer that it would otherwise have done.

317
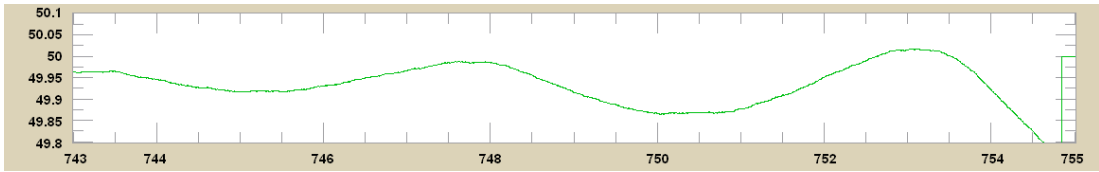
Figure 6-61 : 11.4s trip time with 5% frequency and 40% voltage droop; Frequency (Hz)
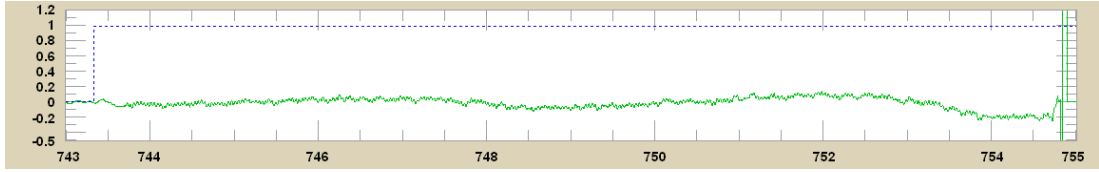
Figure 6-62 : 11.4s trip time with 5% frequency and 40% voltage droop; ROCOF (Hz/s) and instant of LOM inception (blue)
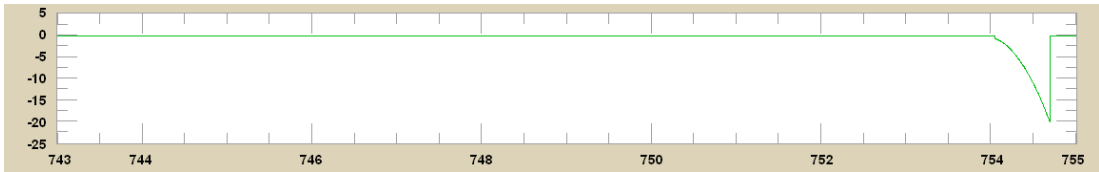
Figure 6-63 : 11.4s trip time with 5% frequency and 40% voltage droop; Phase Offset (degrees)
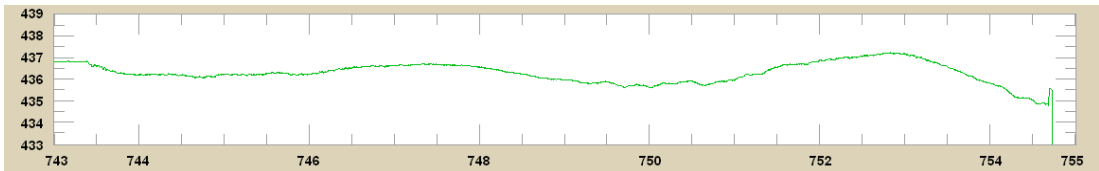
Figure 6-64 : 11.4s trip time with 5% frequency and 40% voltage droop; Voltage (+ve sequence fundamental, line-line)
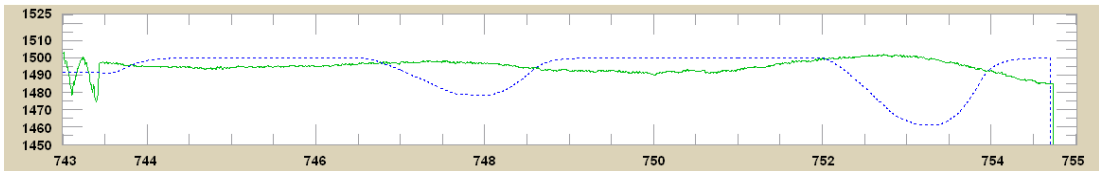
Figure 6-65 : 11.4s trip time with 5% frequency and 40% voltage droop; DG Real power output (solid) and target (dashed)
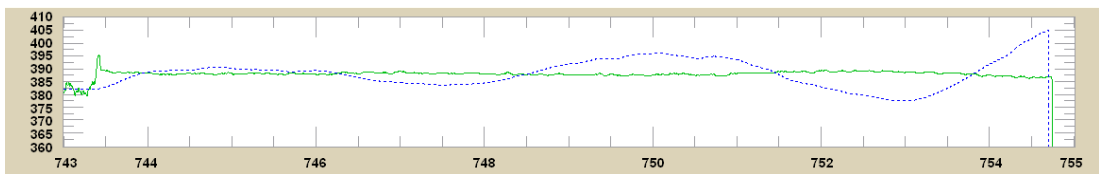
Figure 6-66 : 11.4s trip time with 5% frequency and 40% voltage droop; DG Reactive power output (solid) and target (dashed)

318

6.3.7 DG control strategy 4 – Real power match and 10% voltage droop, guaranteed LOM NDZ avoidance using non-zero VAR exchange

The performance of control strategy 4 (see section 6.3.1.4) in combination with the POR (designed in section 6.1.1) was tested many times, with some different combinations of settings relevant to traditional and microgrid applications.

These tests show that the combination of strategy 4 with the use of the POR is a good way of controlling the DG unit, and guaranteeing LOM detection within a prescribed timeframe. The target timeframe in this case is 2 seconds (IEEE 1547, 2003). To achieve the target timeframe, the ROCOF trigger setting and power flow thresholds (see section 6.3.1.4) must be set appropriately.

The first set of tests use a more conventional ROCOF trigger setting of 0.2 Hz/s for the POR (together with the trip setting of 20° which was used for all tests). In this case, the DG control strategy was set to avoid a VAR exchange with the parent network of smaller than ±0.05pu; with 1pu reactive power being 2000 VAR, this equates to ±100 VAR. This configuration represents a good solution for operating a microgrid in grid-connected or islanded mode when its parent network is the national grid, which has relatively good frequency stability.

The second and third sets of tests use ROCOF trigger settings of 1 Hz/s and 0.5 Hz/s. This significantly lowers the sensitivity of the POR. The reason that this step may need to be taken is to accommodate grid-connection to a parent network significantly smaller than the national grid. For example, as Table 2-1 showed, a 100kVA network will regularly achieve ROCOF rates of up to 0.4 Hz/s simply with 3kW load steps. Larger load steps would produce larger ROCOF rates, even though they may only be transitory events until prime movers respond. In the second and third sets of tests, to mitigate the de-sensitisation of the LOM triggering, the control strategy was adjusted to avoid a VAR exchange with the parent network of smaller than ±0.1pu (±200 VAR). This tends to initially de-stabilise the local power system twice as much as the ±0.05pu offset used in the first tests.

The procedure for these tests is slightly different than for the previous sections. The

active power balance is achieved automatically by the automatic control algorithm[1]. The AVR droop control imports or exports VARs in a manner which provides voltage support. During these tests, the parent network voltage was about 430V RMS line-line. The degree of reactive power balance was thus changed between the test runs by altering the voltage which the AVR regards as its nominal target. This alters the generator reactive power output via the 10% voltage droop slope. Only the novel non-detection-zone (NDZ) avoidance algorithm prevented a more exact balance being achieved, which would have led to some long trip times similar to those of Figure 6-54.

6.3.7.1 Detection times: 0.2 Hz/s ROCOF trigger setting, >±0.05pu VAR exchange

16 test runs were completed, with active power exactly balanced. As hoped, the detection times using these settings were all less than 2 seconds. Figure 6-67 shows the results, tabulated as detection times versus VAR import from the parent network. The VAR import/export from the parent network is never smaller than 0.05 pu due to the action of the LOM NDZ avoidance strategy. Clearly, detection time peaks at about 1.5 seconds for the most closely matched events, but decreases if the VAR exchange is bigger than 0.05 pu. The scatter of detection times between 1.1 and 1.5 seconds, for the same 0.05pu VAR import/export conditions, is due to tiny random fluctuations in the actual hardware test conditions (frequencies, power flows, throttle responses, phase angles, and measurements) which lead to relatively larger detection time variations due to the unstable nature of the system (see Figure 6-55 to Figure 6-60). This effect is similar to that of the difficulty in making accurate weather forecasts due to the way that small local variations can lead to large effects on wide areas over the following days.

---

[1] Taking into account the additional 20W of parasitic trapped load which will be acquired by the microgrid when the loss-of-mains (LOM) event occurs
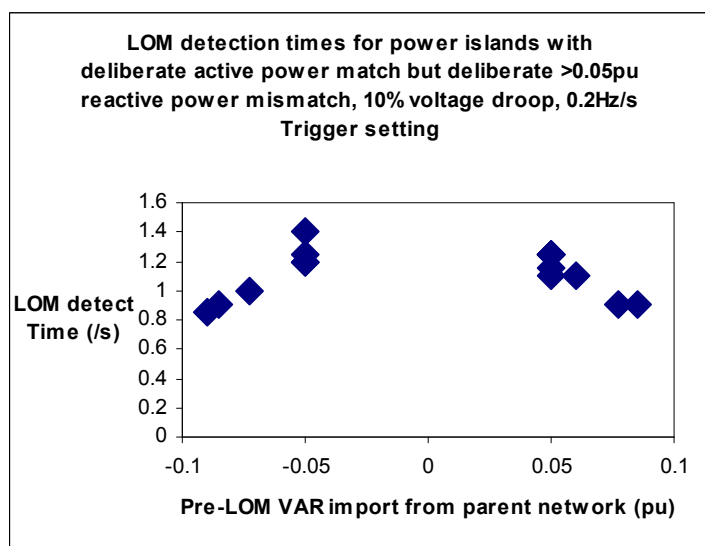
Figure 6-67 : LOM detection times for power islands with deliberate active power match but deliberate >0.05pu reactive power mismatch, 10% voltage droop, 0.2Hz/s Trigger setting

These detection times are all acceptable, being below the 2 second limit given by IEEE 1547 (IEEE, 2003) and quicker than an expected auto-reclose action. The maximum trip time using the proposed NDZ avoidance strategy (strategy 4) is about 1.5 seconds, compared to trip times up to 11 seconds which were possible without the NDZ avoidance strategy. The reduction in maximum trip time is achieved because the NDZ has been correctly avoided by strategy 4. This combination of control strategy and relay settings is therefore appropriate and robust for use within scenarios where the parent network is very large, and expected ROCOF rates due to normal network frequency deviations are less than 0.2 Hz/s, and the total local inertia (DG plus loads) is approximately H=1 pu.

6.3.7.2 Detection times: 1.0 Hz/s ROCOF trigger setting, >±0.1pu VAR exchange

In these tests, the ROCOF trigger setting was set very wide, at 1.0 Hz/s. This would allow, for example, a 6-7kW load step within a 100kVA parent network, to which an even smaller microgrid was grid-connected, without causing a LOM trip (see Table 2-1). 22 test runs were completed. The results are shown in Figure 6-68. The control strategy in this case is set to avoid VAR exchanges of less than 0.1 pu. The test runs which have VAR exchange magnitudes larger than 0.1 pu cause LOM to be detected in less than about 2 seconds. However, some of the test runs which only had a 0.1 pu VAR exchange before the LOM occurred, produced detection times of up to 8 seconds. In some of these runs, the POR was never triggered at the 1 Hz/s level, and therefore could not trip. In these cases the final trip was due to under-frequency or over-frequency at the 47 and 52 Hz levels, which

is not ideal. These trip times are unacceptably long, considering the risk of auto-reclose action and the potential stress damage to contactors or machines.
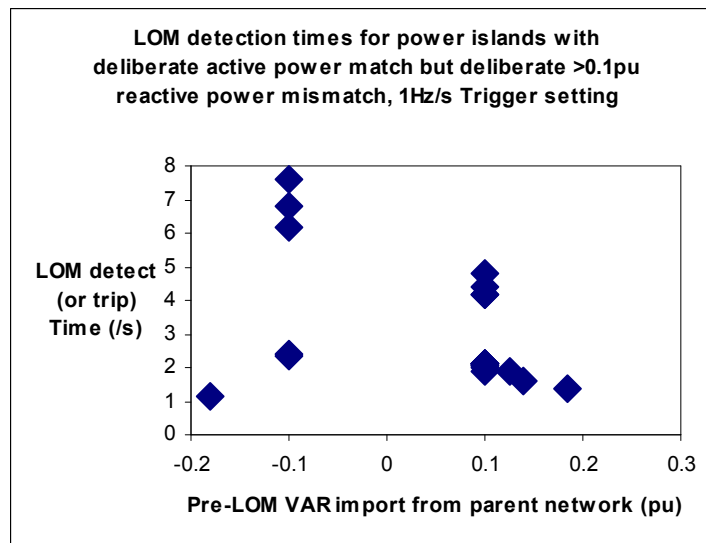
Figure 6-68 : LOM detection times for power islands with deliberate active power match but deliberate >0.1pu reactive power mismatch, 1Hz/s Trigger setting

This indicates, that at a trigger setting of 1 Hz/s, the guaranteed minimum VAR exchange with the parent network should be larger then 0.1 pu, in order to avoid the NDZ and guarantee tripping within 2 seconds.

6.3.7.3 Detection times: 0.5 Hz/s ROCOF trigger setting, >±0.1pu VAR exchange

Following on from the above test, the minimum VAR exchange with the parent network was held at ±0.1 pu, but the ROCOF trigger setting for the POR was reduced to 0.5Hz/s. This would allow, for example, a 7.5kW load step within a 200kVA parent network, to which an even smaller microgrid was grid-connected, without causing a LOM trip (extrapolated from the 100kVA entry in Table 2-1). If the parent network rating was at least 1MVA (with a rotating generator), then routine load steps of up to about 50kW could be made without tripping off smaller microgrids, by exceeding the 0.5 Hz/s ROCOF trigger level (again from Table 2-1). These maximum allowed load steps are 4-5% of the network capacity, with an assumed total inertia (generators and loads) of H=2pu.

So, assuming that genuine ROCOF levels within the parent network are almost always less than 0.5Hz/s, using a minimum VAR exchange of ±0.1 pu, the LOM detection times are as shown below.
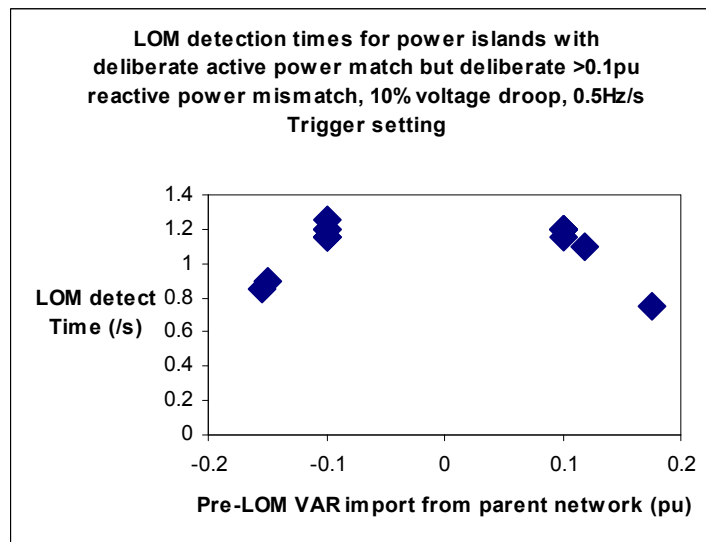
Figure 6-69 : LOM detection times for power islands with deliberate active power match but deliberate >0.1pu reactive power mismatch, 0.5Hz/s Trigger setting

15 test runs are shown in Figure 6-69, although some of the datapoints with trip times of about 1.2 seconds overlie each other, giving the impression that only 9 test runs were completed. The longest detection time was 1.25 seconds, an acceptable figure, below the 2 second limit given by IEEE 1547 (IEEE, 2003) and quicker than an expected auto-reclose action. The detection times where the VAR exchange was higher than 0.1 pu are shorter. This combination of control strategy and relay settings is therefore appropriate for use within microgrid scenarios where the parent network is significantly smaller than the national grid, but the maximum step load change is restricted to approximately 4-5% of the parent network capacity, the parent network system inertia is approximately 2pu, and the local system inertia (DG plus loads) is approximately H=1 pu..

It is also interesting to note that detection times of about 1.2 seconds in Figure 6-69 compare to similar detection times in Figure 6-67. This is after a 2.5x increase in the ROCOF trigger threshold, from 0.2 Hz/s to 0.5 Hz/s, and a 2x increase in the VAR exchange. This demonstrates that even small VAR exchanges help to significantly destabilise the grid-connected power network when it experiences a LOM event, and that the magnitude of the de-stabilising effect is at least linearly related to the VAR flow.

The logs from the longest detection event of Figure 6-69 are shown below. The effect of the deliberate Pre-LOM 0.1 pu reactive power exchange with the parent network can be seen in Figure 6-75, as the generator reactive power output drops by 0.1 pu (200 VAR) when the LOM event occurs. Subsequent to the LOM occurring, the voltage within the

power island then rises as the generator controls try (but fail) to re-attain the target VAR flow. The rising voltage causes an increase in real power demand from the resistive loads, which drags frequency downwards. Due to the real power matching algorithm of control strategy 4, which is used in this test, the generator real power target is also gradually increased. This is caused by integral control action of any actual power flow to the parent network (which in this case is just slightly off zero due unaccounted-for trapped parasitic load). This has a mitigating effect on the rate of the frequency excursion but the DG control response is lagged due to the control filters and integral action, hence the frequency excursion is still negative. There is no frequency restoration control or frequency droop applied, and so the resulting system is unstable both in terms of frequency and voltage.
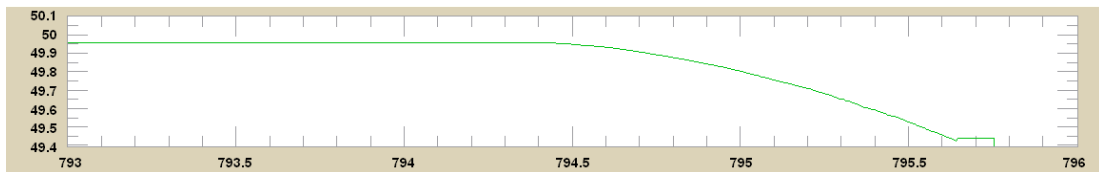
Figure 6-70 : 1.25s trip time with automatic NDZ avoidance; Frequency (Hz)
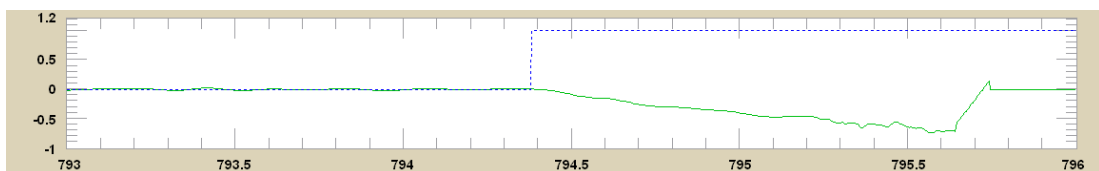
Figure 6-71 : 1.25s trip time with automatic NDZ avoidance; ROCOF (Hz/s) and instant of LOM inception (blue)
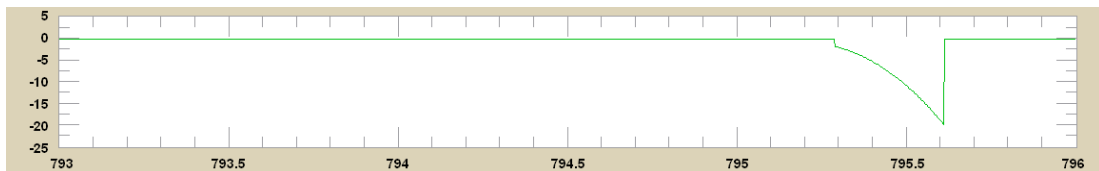
Figure 6-72 : 1.25s trip time with automatic NDZ avoidance; Phase Offset (degrees)
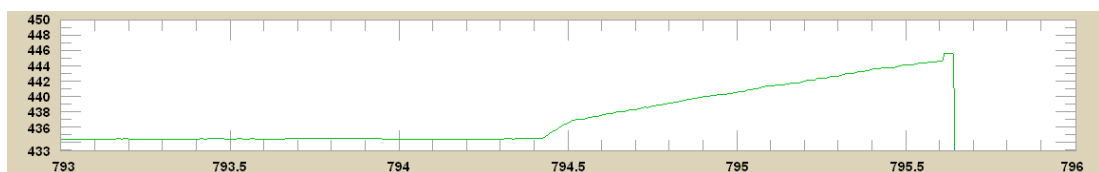
Figure 6-73 : 1.25s trip time with automatic NDZ avoidance; Voltage (+ve sequence fundamental, line-line)
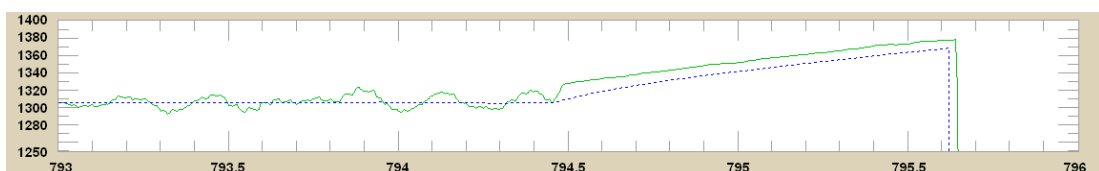
Figure 6-74 : 1.25s trip time with automatic NDZ avoidance; DG Real power output (solid) and target (dashed)

324
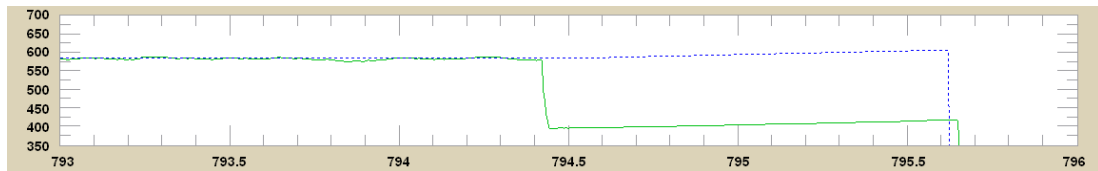
Figure 6-75 : 1.25s trip time with automatic NDZ avoidance; DG Reactive power output (solid) and target (dashed)

6.3.8 Summary findings and further work opportunities resulting from hardware testing

The primary conclusions from this section are:-

- 4 different strategies (with active and reactive power set-points) have been compared for grid-connected generator control, to see which ones resulted in the most effective detection of loss-of-mains.

- The two worst grid-connected control algorithms examined have fixed real and reactive power outputs (zero droop) or active matching of both real and reactive power within the local power system. Examples of sustained non-detection of loss-of-mains were created in the laboratory, using both these control methods.

- Adding 5% frequency droop and 10-40% (voltage) reactive power droop controls (containing appropriate low-pass filtering and integral control within the control algorithms, and lags within the prime mover hardware and generator field), creates an unstable system if the local power system is accidentally placed in islanded mode. However, detection times of between 5 and 12 seconds can still be demonstrated, by creating very close matches of DG output to local load demand (both real and reactive), and placing the local power system with the non-detection-zone (NDZ) of the loss-of-mains detection relay.

- Changing the active power control to an active balancing of real power between local loads and generation, in conjunction with a 10% (voltage) reactive power droop control, also creates an unstable system in islanded mode, although long detection times can still result.

- Further, a novel algorithm can be added which insists on a non-zero (at least ±0.05pu to ±0.1pu) reactive power exchange with the parent network, if the real power exchange is almost zero. This can be used to reduce the maximum detection time from >10 seconds to <2 seconds which is an acceptable figure even in systems with auto-reclose breakers. The active and reactive power thresholds can be set appropriately to meet either the 2 second (IEE 1547 (IEEE, 2003)) or 1 second (ETR 113 (ENA, 1995)) requirements for LOM detection.

325

- Suitable settings for the relay trigger threshold depend upon the size, and inertia, of the parent network to which a microgrid is connected, the maximum permitted step load within the parent network, and the inertia of the local DG/loads. Table 2-1 in conjunction with the results of sections 6.3.7.1 to 6.3.7.3 can be used as a guide for suitable trigger levels, scaled proportionately if necessary.

- If the parent network is a large network such as the national grid, and the local DG/loads have a total inertia of approximately H=1pu, then suitable settings for the relay are a 0.2 Hz/s trigger threshold and a 20° trip threshold (temporarily widening to 100° during balanced or unbalanced faults, as described in section 6.1). In this case either the real or reactive power exchange with the parent network should be kept at a magnitude >0.05pu to guarantee detection of LOM within 2 seconds.

- If the parent network is of the order of 1MVA capacity, with inertia of approximately H=2pu and a maximum load step of 50kW, and the local DG/loads have a total inertia of approximately H=1pu, then suitable settings for the relay are of the order of a 0.5 Hz/s trigger and a 20° trip signal (temporarily widening to 100° during balanced or unbalanced faults, as described in section 6.1). In this case either the real or reactive power exchange with the parent network should be kept at a magnitude >0.1pu to guarantee detection of LOM within 2 seconds. The trigger threshold can be widened further from 0.5 Hz/s if required, to avoid spurious tripping when routine load/generator switching causes ROCOF events larger than 0.5 Hz/s. In this case, the minimum real/reactive power imbalance should also be increased proportionately from 0.1pu, to guarantee LOM detection within 2 seconds.

- Detection times tend to be significantly reduced where inverter-connected distributed generators are installed, due to the low "inertia" of the PLL. However, this statement is dependent upon inverter software design and also any inertia of connected local loads. These are installation dependent. For example, inverters might in future contain artificial inertia within the PLLs to aid fault ride-through etc..

Further work opportunities include:-

- Enhancing an opportunity from section 6.1.5: design and test an algorithm to automatically adapt not only the ROCOF triggering level $R_{Trigger}$ for the POR, but also the minimum allowed real/reactive power imbalances P_t & Q_t within the control algorithm, for different scenarios/size of grid during islanded operations. The adaptive algorithm would need to monitor (at least) frequency deviations due

to "normal" network behaviour.

- To add significant spinning loads with inertia, to simulate an industrial environment, and verify whether the detection times are changed significantly, for the same control algorithm settings and relay settings.

- Du (2005) and Ye (2006) propose the addition of positive feedback to the P and Q controls (throttle and field) based upon differential filtered or bandpass filtered measurements of the measured system voltage and frequency. These systems as presented are undesirable since they reduce or remove the droop controls to enable network support. If the droop controls used in this thesis could be combined with the positive feedback differential terms from Du (2005) or Ye (2006), faster detection times might result for the same small P or Q imbalances.

- To examine more combinations of ROCOF trigger threshold, minimum power exchange thresholds, and DG/load inertias, to derive a theoretical or empirical equation tying these parameters together with a given maximum LOM detection time limit.

6.4   References for chapter 6

ADI. (2008). Applied Dynamics International. [Online]. [Accessed 14/03/2008]. < http://www.adi.com/>

Affonso, C. et al. (2005). Performance of ROCOF relays for embedded generation applications. IEE Proceedings on transmission and distribution, vol. 152 no.1, pp. 109-114.

Areva T&D. (2007). Network Protection and Automation Guide. [Online]. [Accessed 26/11/2007]. <http://www.areva-td.com/scripts/solutions/publigen/content/templates/show.asp?P=1228&L=US> Chapter 14

Dysko, A., Burt, G. & Bugdal, R. (2006). Novel Protection Methods for Active Distribution Networks with High Penetrations of Distributed Generation – Year 2 Report. DTI/CDGSEE/TR/2006-7. [Online]. [Accessed 14/03/2008]. <http://www.sedg.ac.uk/DGSEE Protection Project - Year 2 Report.pdf>

Dysko, A. et al. (2007). Reducing unnecessary disconnection of renewable generation from the power system. IET Renewable power generation, vol. 1 no. 1, pp 41-48.

Dysko, A. et al. (2007b). UK distribution system protection issues. IET Generation, transmission and distribution, vol. 1 no. 4, pp 679-687.

Du, P., Nelson, J. & Ye, Z. (2005). Active anti-islanding schemes for synchronous-machine-based distributed generators. IEE Proceedings on generation, transmission and distribution, vol. 152 no. 5, pp 597-606.

ENA. (1991). ER G59/1. Recommendations for the connection of embedded generating

plant to the public electricity suppliers' distribution system.

ENA (1995). ETR 113. Notes for guidance of the protection of embedded generation plant up to 5MW for operation in parallel with public electricity suppliers' distribution systems.

Freitas, W. et al. (2005). Comparative analysis between ROCOF and Vector Surge relays for distributed generation applications. IEEE Transactions on power delivery, vol. 20 no. 2, pp. 1315-1323.

Huang, S. & Pai, F. (2001). Design and operation of grid-connected photovoltaic system with power-factor control and active islanding detection. IEE Proceedings on transmission and distribution, vol. 148 no. 2, pp 243-250.

IEEE. (2003). IEEE 1547, IEEE Standard for interconnecting distributed resources with electric power systems.

Lokov, G. & Finstad, R. (2005). Active anti-islanding system for an electric power plant and his method. European Patent no. EP 1 574 672 A2.

Mullane, A., Bryans, G. & O'Malley, M. (2005). Kinetic energy and frequency response comparisons for renewable generation systems. Conference on future power systems. Paper O08-01.

Roscoe, A. (2008). Power flow management to avoid the non-detection of loss of mains. UK patent application 0810512.4 filed 10 June 2008.

RTDS. (2008). RTDS technologies. [Online]. [Accessed 14/03/2008]. <http://www.rtds.com/index.html>

SEDG. (2008). United Kingdom Generic Distribution System (UKGDS). [Online]. [Accessed 14/03/2008]. <http://www.sedg.ac.uk>

Sumner, M., Palethorpe, B. & Thomas, D. (2004a). Impedance measurement for improved power quality - Part 1: The measurement technique. IEEE Transactions on power delivery, vol. 19 issue 3 pp 1442-1448.

Sumner, M., Palethorpe, B. & Thomas, D. (2004b). Impedance measurement for improved power quality - Part 2: A new technique for stand-alone active shunt filter control. IEEE Transactions on power delivery, vol. 19 issue 3 pp 1457-1463.

Timbus, A. et al. (2004). Online grid measurement and ENS detection for PV inverter running on highly inductive grid. IEEE power electronics letters, vol. 2 no. 3, pp 77-82.

Vieira, C. et al. (2006a). Performance of frequency relays for distributed generation protection. IEEE Transactions of power delivery, vol. 21 no. 3, pp 1120-1127.

Vieira, C. et al. (2006b). Efficient coordination of ROCOF and frequency relays for distribution generation protection by using the application region. IEEE Transactions of power delivery, vol. 21 no. 4, pp 1878-1884.

Vieira, C. et al. (2006c). Formulas for predicting the dynamic performance of ROCOF relays for embedded generation applications. IEE Proceedings on generation transmission and distribution, vol. 153 no. 4, pp 399-406

Wall, S. (2004). Detection of islanded behaviour and anti-islanding protection of a generator in grid-connected mode. US Patent no. US 2004/0178641 A1.

Ye, Z. et al. (2004). Evaluation of anti-islanding schemes based on nondetection zone concept. IEEE Transactions on power electronics, vol. 19 no. 5. pp 1171-1176.

Ye, Z. et al. (2006). Anti-islanding protection systems for synchronous machine based distributed generators. US Patent no. US2006/0082936 A1

7   Conclusions

The work of this thesis enables an increase in the security of supply within microgrid scenarios using distributed generation. This is accomplished by addressing two identified gaps in established knowledge.

1) Accurate, timely measurement of amplitude/phase/frequency with low frame rates within power systems experiencing poor power quality. The achievement of this goal allows many simultaneous measurement and control functions to be integrated on a single cheap microcontroller at an appropriate frame rate, suitable for ubiquitous deployment with small-scale generation at low incremental cost. The resulting distributed control algorithms can be used to enhance security of supply by offering the possibility of network reconfiguration, frequency support, ancillary services provision, and/or deliberate islanding.

2) Reliable, timely detection of Loss-Of-Mains (LOM) when local real power generation is deliberately balanced to local real power demand, while avoiding spurious (nuisance) tripping due to switching, noise, harmonics, and network faults. The ability to detect LOM reliably, despite a match of active power within the microgrid, allows a microgrid controller to switch quickly to islanded mode with a minimal frequency and voltage excursion following a LOM event, which increases the security of supply at the local level.

To accomplish the main overall goals, a comprehensive system-level study of the requirements to be met and the potential scenarios has been carried out. The requirements to be met are driven by required control action times (latencies) and required measurement accuracies for control and relaying purposes. A large emphasis is placed upon measurement ripple magnitude, since measurement ripple can cause power system oscillations when fed back into the system via control action. The system requirement study brings together information from a number of sources. Some data is directly available from applicable standards, but much is calculated from potential worst-case scenarios in non-standard applications such as rural, islanded, battleground or disaster-relief scenarios.

The first conclusion of the requirement study is that rates of change of frequency within microgrid scenarios can regularly be much higher than seen normally within the UK power grid. This presents problems for many previously published frequency measurement algorithms. A second major conclusion is that harmonic distortion on voltages could in theory reach 53% THD in weak power systems containing predominantly lighting and

computer/IT loads. Operating any power system in such a condition is undesirable for any length of time, but this upper limit is used throughout this thesis as a test case for candidate measurement algorithms to ensure robust operation. The potential magnitudes of other "influence qualities" such as inter-harmonics, flicker, noise, unbalance etc. are also calculated and used to generate suitable test waveforms. These test waveforms are invaluable for verification of the performance of the measurement algorithms proposed in this thesis, and could also be used for other projects to test other measurement/control algorithms.

To measure AC system parameters, the use of exact-time averaging is proposed by this thesis as the single most powerful and applicable building block. Only one previous work concerning AC signal measurements applies such a technique, and in that work the technique is simply used as a block and not analysed. This thesis performs an in-depth analysis of the properties of such an algorithm. The practical difficulty lies in averaging a signal over an exact timeframe which may not be an integer multiple of the sample interval. A version of an algorithm to carry out such averaging is included within the MATLAB SimPowerSystems blockset, but several significant deficiencies have been noted and overcome. The improvements made during this thesis include speed increases, improvements to the latency and coherency of the algorithm output, extension to 2nd order interpolation, and modification of the code to allow robust operation over long periods of time in embedded processing applications. Without such improvements, the MATLAB exact-time averaging algorithm was significantly limited in application scope. With these improvements, the algorithm can now be used as a robust building block for AC signal measurements in embedded target processors.

A major contribution of this thesis is the development and demonstration of simple but effective algorithms using repeated application of this exact-time averaging block. Two key properties of this exact-time averaging algorithm are identified within this thesis, which, now being fully understood, help to define optimum architectures for measurement algorithms.

The first property is that the output of such an averaging block (for a perfect sinusoidal input and over a time period of exactly 1 cycle) exhibits ripple due to interpolation error and the finite sample interval, but that the frequency of the ripple is predictable. Typically, when measuring a Fourier correlation the input signal is at frequency f, and the Fourier correlation products are therefore at frequency $2f$. Averaging over one cycle period then leads to interpolation error at a frequency of $2f$. This can be almost

completely removed by a further exact-time average over ½ a cycle period. Thus, a single cycle Fourier correlation followed by a ½ cycle average is shown in this thesis to be an extremely simple but powerful measurement architecture to allow low sample rate operation down to 10 samples per cycle, with performance surpassing that of more complex 2nd harmonic cancellation techniques.

The second property highlighted by this thesis is that cascaded average filters provide excellent attenuation of Gaussian noise, with performance considerably exceeding that of low-pass filters with equivalent latencies. The cascading of two 2-cycle averaging filters, via the convolution of the impulse responses and the pole-zero placements, also provides much better noise reduction than does a single 4-cycle averaging filter.

Combining the new knowledge of these two important properties, this thesis shows how the exact-time averaging block can be used to build a number of adapted and entirely novel stages, and that these stages can be cascaded in optimum novel configurations to create extremely effective measurements of AC signals. The measurement latency can be traded off against minimisation of ripple and noise, under adverse conditions of harmonics, inter-harmonics, noise, ADC quantisation, and other interfering signals. This thesis shows how to make the optimum measurements with latencies (within the digital computations) of ½, 1, 1½, and more than 1½ cycles, with increasing performance as allowed latency increases.

The largest single problem with measurements at low sample rates is identified in this thesis as that of aliased harmonics (predominantly the 9th and 11th) causing low frequency ripple at the measurement output. This is found to be a hurdle to making adequately ripple-free measurements at a main frame rate of 10 samples per cycle, in the cases of worst harmonic distortion. To comply with the system-level requirements, two solutions to this problem are proposed together. Firstly, a novel adaptive ripple-remover is designed and implemented. This measures the frequency of any sub-harmonic ripple and removes it as far as possible, using exact-time averaging over reasonably long timeframes as allowed within the measurement latency. An important feature of this algorithm is the ability to automatically bypass itself during transients, thus providing faster measurements at the expense of increased noise/ripple when appropriate. This is extremely valuable since it can be used within algorithms which are used both for control (where low ripple is of primary concern) and also for relaying (where response time is of primary concern). Secondly, to achieve the lowest ripple performance (±0.001pu ripple on voltage measurements with 28% THD), it is also necessary to oversample the ADCs at 3kSa/s (6x

oversampling) and apply very small 6-tap FIR filters at this higher frame rate, before downsampling to 500 Sa/s (10 samples per cycle) for the main measurement algorithm. However, such an oversampled measurement architecture does not add expense to measurement hardware because it can be realised on existing microcontroller platforms such as the Infineon TC1796.

A general measurement architecture proposed by this thesis is shown in Figure 4-42. This architecture, which is fully implemented and tested within this thesis, allows "Class A" measurement accuracies to be achieved but at much lower sample rates and measurement latencies than traditionally used.

To measure frequency effectively, the same architecture is used with embellishments. More than 8 candidate solutions were compared, from which the final solution has been developed and selected. The best solution is found to be a hybrid of a 3-phase Clarke-transform based measurement and a novel Frequency Locked Loop (FLL) algorithm which measures both frequency and voltage (amplitudes & phases) of a 3-phase voltage set. The hybrid combines the best properties of each algorithm. The Clarke transform measurement is very fast settling due to the nominally constant speed of rotation of the **AB** vector, but does not function well during two-phase faults due to collapse of the **AB** vector trajectory to a straight line. The FLL is slower settling but can tolerate two-phase faults and large levels of unbalance without detriment to measurement accuracy/ripple. Coupling the two algorithms together with the appropriate averaging stages and decision processes is a significant achievement and provides an ideal solution. It is shown to provide significantly better response than algorithms based upon zero crossings, phased locked loops (PLLs) or other previously published techniques. Although the proposed solution is not a PLL, many of the concepts and algorithmic blocks could be applied to PLL applications to good advantage.

The Clarke-FLL hybrid algorithm also includes code which provides the following additional features:-
- Self-checking for validity of frequency measurement output
- Initial fast-settling of frequency measurement within 2 cycles (40ms) following signal application.
- Ride-through of frequency measurement for configurable time periods during brief 100% three-phase voltage dips, to allow the maximum potential for riding through system faults without tripping generators or loads unnecessarily.

None of these features have been seen before within published algorithms.

To prove the robustness and applicability of the new algorithms, they have been implemented and benchmarked on two real-time hardware platforms: the Infineon TC1796 microcontroller and the Real-Time-Station from Applied Dynamics International. Use of the algorithms over extended periods of time has proved their computational robustness in the real-time environment, which requires much more carefully constructed software than required by simple simulation exercises. The algorithms have been incorporated into several applications used in the laboratory at Strathclyde, including a microgrid management control agent (running at 500 samples per second) and a power quality meter (running at 1500 samples per second).

Thorough benchmarking exercises were also conducted to measure the breakdown of the algorithm execution times; these are presented in Appendix G. Such data is rarely available and even more rarely published, but is extremely valuable as a tool for speed improvement and higher-level system design. The data was used during this thesis to incrementally improve the speed of several key algorithmic blocks and allowed an overall execution time reduction of 25 to 50% for the major algorithms. Re-use of calculations and minimisation of trigonometric function evaluation within the analysis blocks also contributes to the small execution time. The total execution time for measurement of 3-phase voltages and currents at a node, with full sequence analysis and power flow analysis, is 156μs on the TC1796 microcontroller, less than 8% of a 2000μs frame time at 500 samples per second. Importantly, this leaves the remaining 1844μs frame time available for other generator/microgrid measurement and control functions

In addition to the creation and testing of the fundamental measurement algorithms, an algorithm has also been realised for a new type of passive loss-of-mains detection relay, called a Phase Offset Relay (POR). This is the first implementation of such a relay in a robust version suitable for deployment on a real-time target. This relay is independent of generator type, and uses local voltage measurements only. The relay uses calculations of perceived phase offset, relative to a parent network.

It is shown that this relay can be used to successfully detect loss-of-mains with generation-load imbalances of only 2.5% (real or reactive) within the 2 seconds allowed by IEEE 1547, even for power systems containing synchronous generators, using a trigger setting of 0.15-0.2Hz/s and a trip setting of 20°. This is comparable with the best available ROCOF relays using ROCOF trip settings of 0.15-0.2Hz. However, the main aim of this new relay algorithm is to bring the following additional benefits:-

- Much lower risk of spurious tripping due to noise, due to the double-integration (averaging) stages involved in the conversion from the ROCOF figure to first frequency and then phase.

- Much lower risk of spurious tripping due to extended trip times (up to that allowed by the system operator) when ROCOF only just exceeds the trigger threshold but the phase offset accumulates slowly. This discriminates genuine LOM events against normal load steps much better than a ROCOF relay.

- Much lower risk of spurious tripping due to post-fault power system oscillations, by the design and implementation of a novel new algorithm which dynamically adjusts the trip setting during and immediately subsequent to close-in faults. The relay is shown not to trip during simulated scenarios which cause commercially available and other proposed relays to trip.

The final significant piece of work in this thesis is a new strategy for microgrid management which involves small reactive power flow adjustments. The aim of this is to guarantee detection of a LOM event even when active power is exactly balanced within a microgrid. This deliberate match of active power is desirable as a strategic pre-islanding measure, but without the new reactive power control algorithm, such a power match introduces a high risk of not detecting a LOM event. The new algorithm continually monitors reactive power exchange between a local power system and its parent network (grid), and makes small adjustments to the generator reactive power output if required, such that the non-detection-zone (NDZ) of the loss-of-mains detection relay is always avoided by enough to provide a detectable but not over-large frequency disturbance upon islanding. Allowance is also made for the possibility of trapped load. The algorithm is tested using a real microgrid containing a synchronous generator and real/reactive loads, and shown to be fully effective. A generator stability analysis also shows that standard droop controls can be used, whereas previous published works with similar aims require unconventional droop controls to allow detection of LOM. The use of conventional droop controls is important since it allows frequency and voltage support functions at the same time as avoiding the LOM NDZ. The application of this reactive power control strategy is shown in the laboratory to reduce the worst-case LOM detection time from >10 seconds to <2 seconds, in line with IEEE 1547.

All of the algorithms developed during this thesis have been created to meet specific needs of an integrated microgrid management control system, which executes on a single microcontroller platform. All of the algorithms have been successfully integrated within this control system at 500 Sa/s and perform as described in this thesis, enabling more

reliable operation of the microgrid than was possible prior to the development of these solutions. This has been proved for the frequency/amplitude/phase measurements by trialling several published and novel candidate solutions in the laboratory environment over a period of 4 years, before the algorithms presented in chapters 3 to 5 were finalised. Their measurement latency and noise/ripple behaviour of the Clarke-FLL hybrid surpasses any other method yet tried or published, given the constraints of sample rate laid down in this thesis. This leads to smaller control actions due to noise/ripple, and more stable operation due to the low latency. The effectiveness of the LOM NDZ avoidance strategy has also been proved in the laboratory environment.

The algorithms are coded in a combination of Simulink (MATLAB R14SP1 was used throughout) and C-code (Simulink "S-functions") and are fully robust for long-term real-time deployment. The Simulink "real-time-workshop" and "embedded coder" features have proved to be an effective way of writing error-free code which can be tested in simulation on a PC and then deployed (without code modification) to real-time targets. The combined algorithmic designs and results from this thesis now provide an excellent foundation upon which to build more advanced microgrid control and protection applications.

7.1 Further Work

The measurement algorithms of chapters 3-5 have been tested many times and no deficiencies are currently known. The point of note is that if sample rates higher than 500 Sa/s can be used, then the algorithm performance will be better than described in this thesis. A possible opportunity (depending upon microcontroller capability) is to split the algorithms and to move the initial 1-cycle Fourier correlations onto the peripheral control processor at the oversampled rate of 3 kSa/s (or more), while leaving the main bulk of the algorithms at the much slower sample rate of 500 Sa/s. The oversampled FIR notch filter would be removed. This would further improve the performance of the algorithms and potentially allow the anti-aliasing filter cut-off frequency to be raised, improving the RMS and THD measurements as well as the Fourier fundamental measurement.

Suitable settings for the POR are dependent upon the size and qualities of the parent network. Typical settings for connection to the UK national grid are a ROCOF trigger threshold of 0.15-0.2 Hz/s and a trip threshold of 20°, with an imbalance in real or reactive power of about 0.025 (2.5%) required to guarantee detection of LOM within 2 seconds. For smaller parent networks, the ROCOF trigger threshold needs to be raised to avoid spurious trips due to regularly occurring frequency deviations. A larger reactive power imbalance is then required to guarantee detection within 2 seconds. Exploring the (non) linearity of this relationship, and the practical ROCOF threshold required for different networks, is a significant opportunity for further work. It might be possible to create an algorithm which can continually monitor the magnitude of ROCOF on a power system and automatically set ROCOF trigger thresholds appropriately, to adapt to changing parent network parameters.

The LOM detection tests could be repeated with a significant proportion of high-inertia spinning loads, to verify that the LOM events can still be detected. During the testing of this thesis, only static R&L loads were used due to equipment availability.

Finally, as a modification to the reactive power control algorithm, Du (2005) and Ye (2006) propose the addition of positive feedback to the P and Q controls (throttle and field) based upon differential filtered or bandpass filtered measurements of the measured system voltage and frequency. These systems as presented are undesirable since they include no standard droop controls to enable network support. If the droop controls used in this thesis could be combined with the positive feedback differential terms, faster detection times might result for the same small P or Q imbalances.

8   About the author

Andrew Roscoe received the B.A. degree in Electrical and Information Sciences Tripos at Pembroke College, Cambridge, England in 1991. He was awarded the M.A. degree in 1994, and elected to chartered membership of the IEE in 1996. Andrew worked for GEC Marconi from 1991 to 1995. He was involved in antenna design and calibration, specialising in millimetre-wave systems and solid-state phased array radars. Andrew worked from 1995 to 2004 with Hewlett Packard and subsequently Agilent Technologies, in the field of microwave communication systems, specialising in the design of test and measurement systems for personal mobile and satellite communications. Andrew was awarded an MSc with distinction from the University of Strathclyde in 2004, in the field of "Energy systems and the Environment". Andrew is currently a research fellow in the Institute for Energy and the Environment, Department of Electronic and Electrical Engineering at Strathclyde University, working in the field of distributed generation and active network management. Recent projects include real-time pricing studies, the creation and deployment of microgrid control algorithms at the 100kVA scale, the design/build of a 10kVA 3-phase inverter, new algorithms for the measurement of dynamic system parameters using low sample rates, and loss-of-mains detection strategies.

Appendix A   Lower-level FLL code details

Measure frequency, fundamental amplitude, approximate all-harmonic RMS and approximate THD.
Single phase, using precalculated part A data   Andrew Roscoe, 2007
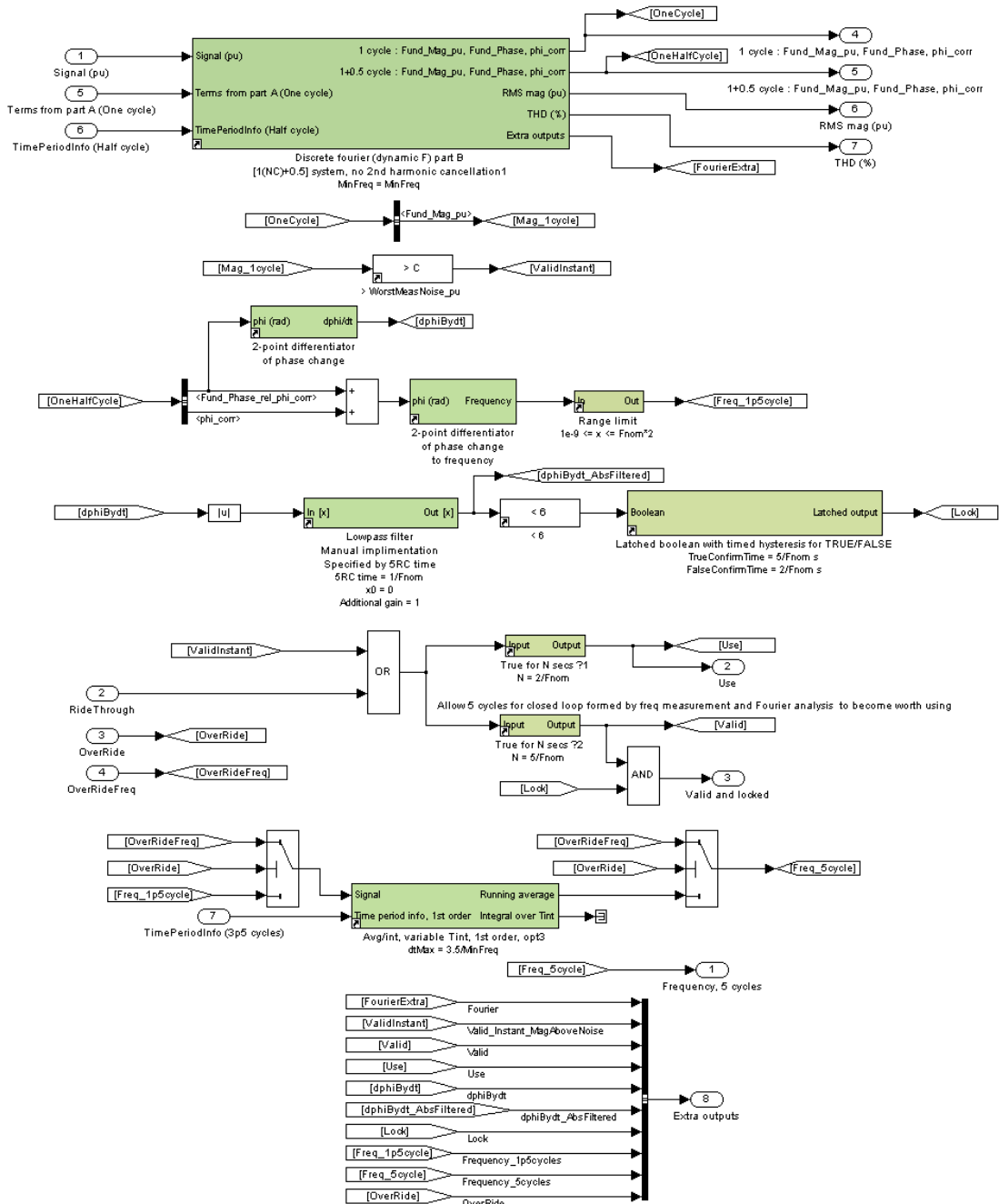
Fig. A-1 : Clarke-FLL hybrid - detail (2); core

Single and two-phase fault ridethrough

Andrew Roscoe, 2007-8

Temporary weighting reductions during single or two-phase disturbances
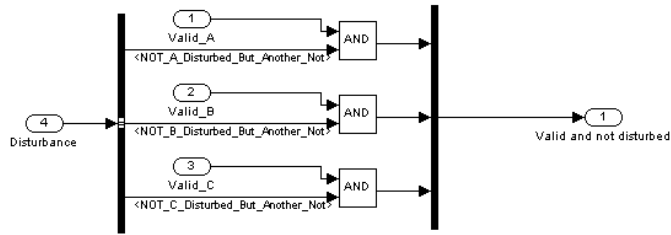
Fig. A-2 : Clarke-FLL hybrid - detail (3); single and two-phase fault ride-through

Calculate frequency from three single-phase measurements,
using weightings determined from their per-unit amplitudes

Andrew Roscoe, 2007
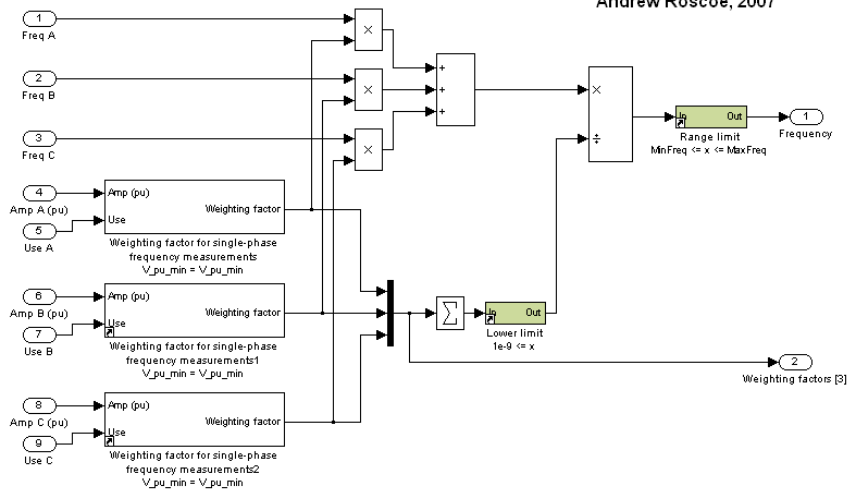
Fig. A-3 : Clarke-FLL hybrid - detail (4); weighted averaging of frequency

Assign a weighting factor for single-phase frequency measurements

Andrew Roscoe, 2007
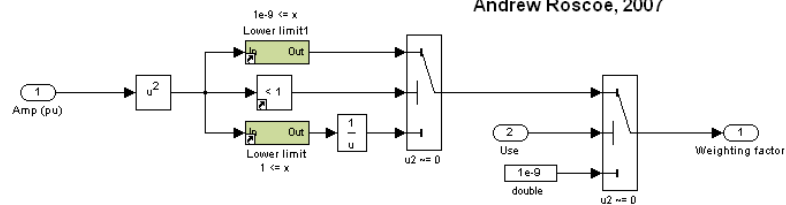
Fig. A-4 : Clarke-FLL hybrid - detail (5); weighted averaging of frequency subroutine

340

Replacement of input with seed ifoutside allowed window

Andrew Roscoe, 2007
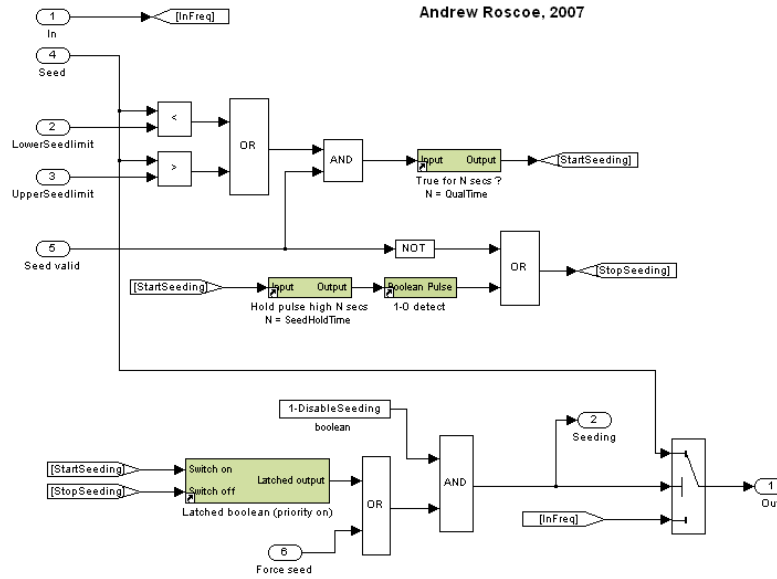
Fig. A-5 : Clarke-FLL hybrid - detail (6); seeding

Determine whether to ride through three-phase disturbances.
Version for systems which can measure with only a single phase up

Andrew Roscoe, 2007-8

This time 2.25/Fnom should match or slightly exceed the take taken to validate the 1.5-cycle measurement.
This way, when a voltage pops back up during ride-through, the 1.5-cycle measurement is validated before the ridethrough action ceases.

Exit conditions:-
BanRideThroughAfterRideThroughTime OR
Any mags are back above the noise for ~one cycle, during Ridethrough

Fig. A-6 : Clarke-FLL hybrid - detail (8); ride-through activation

341

Qualify disturbance on any of three phases.
Only call a "disturbance" if there is a "transient" but
the signal amplitude is about nominal,
and also if the frequency is about nominal.

Andrew Roscoe, 2007

Maximum rate of change of amplitude normally occurs about half cycle after transient, for 1-cycle Fourier base measurement.
Thus, check to see if amplitudes were normal 1/2 a cycle ago, and qualify the transient detection by this result.

This time 2.25/Fnom should match or slightly exceed the take taken to validate the 1.5-cycle measurement.
It also matches the time in the ridethrough exit code

Fig. A-7 : Clarke-FLL hybrid - detail (7); disturbance detection

Signal hold with lookback

Andrew Roscoe, 2007
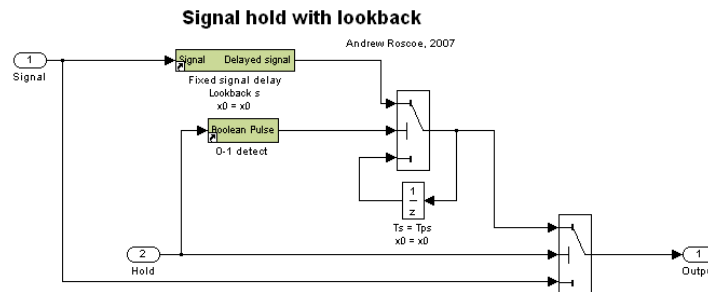
Fig. A-8 : Clarke-FLL hybrid - detail (9); ride-through look-back

342

Appendix B   Useful formulae

B.1   Three-phase power flows

B.1.1   Power, current, voltage and resistance

$$I_p = \frac{P_3}{V_3 \sqrt{3}}$$

<div align="right">{B.1}</div>

$$P_3 = \frac{V_3^2}{R_p}$$

<div align="right">{B.2}</div>

- I_p = RMS Phase current (A)
- P_3 = Three phase power flow (VA)
- V_3 = RMS line-line voltage (V)
- R_p = Resistance, phase to neutral in a balanced star system

B.1.2   Estimation of overhead line voltage required

An estimation of required 3-phase voltage for overhead line connections can be made by the formula:

$$E = k\sqrt{Pl}$$

<div align="right">{B.3}</div>

where:-

- E = 3-phase line voltage (kV)
- P = power to be transmitted (kW)
- l = line length (km)

and k is a coefficient dependent on the impedance (real and reactive) of the line. k varies from about 0.06 for a compensated line, to 0.1 for a compensated line with voltage regulation of 5%.

This formula derives from the limits of phase angle across the transmission line, which should never exceed 90° otherwise the network is at risk of breaking apart. The phase angle is determined by the real power flow and the line inductance, and by the reactive power flow and the line resistance. [Wildi, T (2002). Electrical machines, drives and power systems. Fifth edition. Prentice Hall. ISBN 0 13 098637 2.]

B.1.3   P and Q flows along a transmission line

A   B

$$R + jX$$

$$V_A \angle \delta$$ $$V_B \angle 0$$
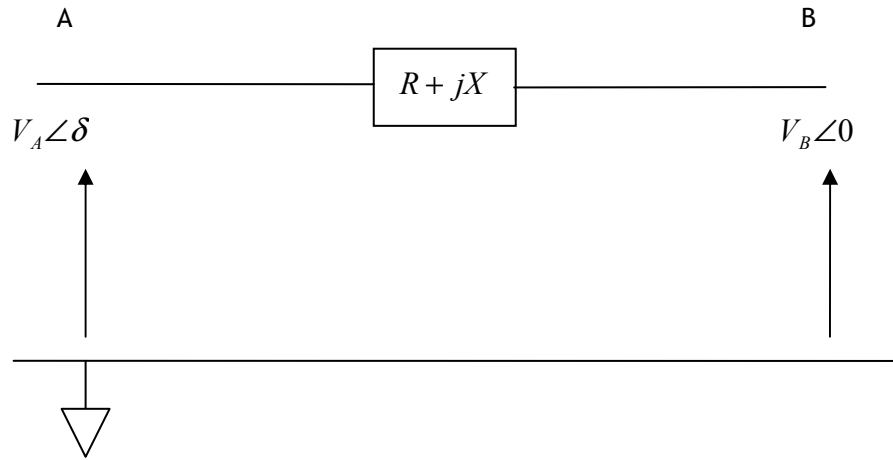
Fig. B-1 : P and Q flows along a transmission line

$$P = \frac{V_A}{R^2 + X^2}\left[XV_B \sin \delta + R(V_A - V_B \cos \delta)\right]$$

$$Q = \frac{V_A}{R^2 + X^2}\left[X(V_A - V_B \cos \delta) - RV_B \sin \delta\right]$$

B.1.4   AB vector trajectory under a single-phase fault

This is easiest analysed by dropping phase A

By (5.3)

$$\begin{bmatrix} A \\ B \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \times \begin{bmatrix} 0 \\ \cos\left(\omega t - \frac{2\pi}{3}\right) \\ \cos\left(\omega t + \frac{2\pi}{3}\right) \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\cos(\omega t) \\ \sin(\omega t) \\ -\frac{1}{3}\cos(\omega t) \end{bmatrix}$$

Other single-phase faults will result in other elliptical traces with aspect ratios of 3:1

B.2   Filters

B.2.1   1st order low-pass filter

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1 + j2\pi fRC}$$ if the filter is made of resistor and capacitor components R & C

The cut-off frequency F_c is set by $\omega_c = 2\pi F_c = 1/RC$

This can also be expressed in the Laplace domain as $\dfrac{1}{1 + \dfrac{s}{\omega_c}}$

Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{\sqrt{1 + \left(\dfrac{f}{F_c}\right)^2}} \angle - \arctan\left(\frac{f}{F_c}\right)$$

{B.4}

Which, for example, gives V_o/V_i as $1/\sqrt{2}$ at $-45°$ if $f=F_c$.

Also, Given $2\pi F_c = 1/RC$, then if we want to define a filter by its "5RC" settling time, i.e. the time taken to settle to within 1% after a step change, then we can determine the appropriate F_c by:

$$F_c = \frac{5}{2\pi \cdot 5RC} = \frac{0.7958}{5RC}$$

{B.5}

B.2.2   Sample time required to accurately model an analogue low-pass filter

A low-pass filter can be approximated by a digital single-pole filter. This can be designed in Simulink, by the following generalised code segment (which can work with significantly more complex filters):-

```
z=sym('z');
% First get Num and Dec in terms of laplace
Num=[1];
Den=[1/(2*pi*LPF_Fc), 1];
H=tf(Num,Den);
Hd=c2d(H,Ts,'zoh');
```

```
[Num_z_cell,Den_z_cell]=tfdata(Hd);
Num_z_0=Num_z_cell{1,1}(2);
Den_z_1=Den_z_cell{1,1}(1);
Den_z_0=Den_z_cell{1,1}(2);


filter_num=Num_z_0
filter_den=z*Den_z_1+Den_z_0
Total_filter=filter_num/filter_den
```

The size of the time step, relative to the cut-off frequency and the actual waveforms input to the filter, determines how accurately the digital filter performance follows that of an analogue filter. This accuracy is important when simulations of analogue filtering hardware are being carried out. For example, in this thesis, a 125Hz low-pass analogue anti-aliasing filter is commonly modelled. Input signals of concern are those up to at least the 40th harmonic of 50Hz, or 2000Hz. To assess what time step is required to accurately represent a 125Hz analogue low-pass filter in the digital domain, a simple MATLAB script can be used.

This script shows that a time-step of approximately 40µs, 1/25000th of a second, or 500Sa/cycle at 50Hz, is required to accurately model the response in gain, to within 0.1dB (for input signals up to 2000Hz). The phase accuracy is only ≈15°. A rule of thumb is therefore that the digital time-step needs to be of the order of 10 times smaller than the period of the highest frequency whose attenuation needs to be accurately modelled through the filter. The plots for 40µs and 100µs time-step filter implementations are shown in Fig. B-2 to Fig. B-5.
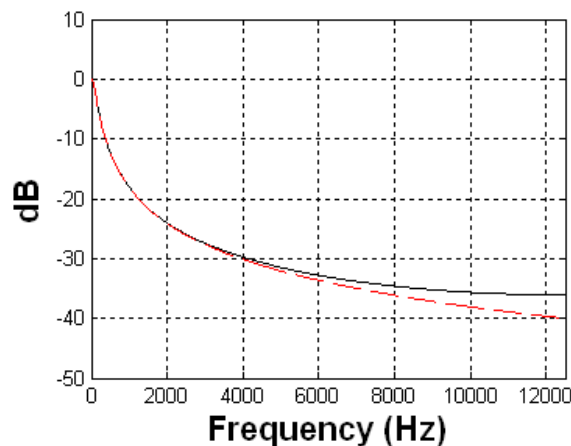
Fig. B-2 : Digital single-pole low-pass filter accuracy (gain), Fc=125Hz, Ts=40µs. Actual (solid) and theoretical (red dashes)

346
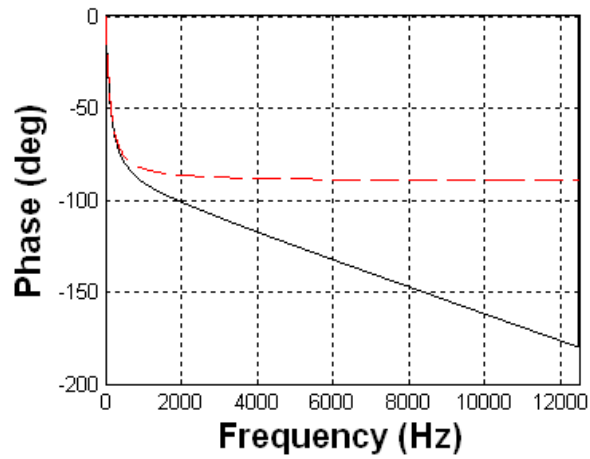
Fig. B-3 : Digital single-pole low-pass filter accuracy (phase), Fc=125Hz, Ts=40μs. Actual (solid) and theoretical (red dashes)
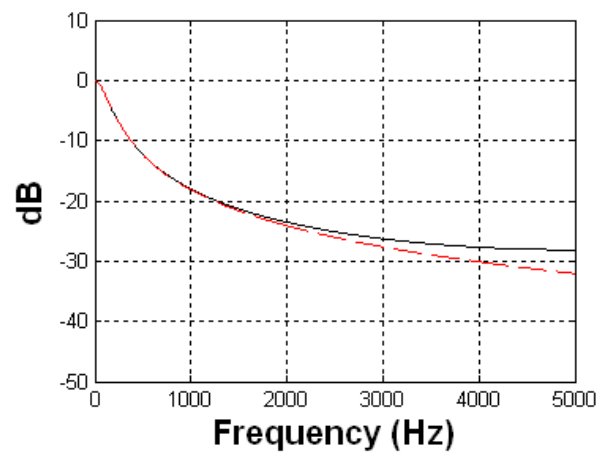
Fig. B-4 : Digital single-pole low-pass filter accuracy (gain), Fc=125Hz, Ts=100μs. Actual (solid) and theoretical (red dashes)
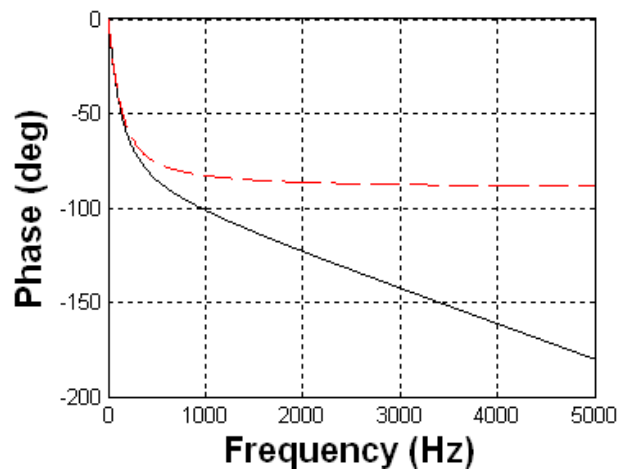
Fig. B-5 : Digital single-pole low-pass filter accuracy (phase), Fc=125Hz, Ts=100μs. Actual (solid) and theoretical (red dashes)

347

B.2.3   1st order high-pass filter

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1+\dfrac{1}{j2\pi fRC}}$$ if the filter is made of resistor and capacitor components R & C

The cut-off frequency F_c is set by $\omega_c = 2\pi F_c = 1/RC$

This can also be expressed in the Laplace domain as $\dfrac{1}{\dfrac{\omega_c}{s}+1}$ or $\dfrac{\dfrac{s}{\omega_c}}{1+\dfrac{s}{\omega_c}}$

Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{\sqrt{1+\left(\dfrac{F_c}{f}\right)^2}} \angle \arctan\left(\dfrac{F_c}{f}\right)$$

{B.6}

Which, for example, gives V_o/V_i as $1/\sqrt{2}$ at $+45°$ if $f=F_c$.

B.2.4   2st order low-pass filter

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1-(2\pi f)^2 LC}$$ if the filter is made of inductor and capacitor components L & C

The resonant frequency F_c is set by $2\pi F_c = 1/\sqrt{LC}$. Hence,

$$\frac{\overline{V_{out}}}{\overline{V_{in}}} = \frac{1}{1-\left(\dfrac{f}{F_c}\right)^2}$$

{B.7}

which has:-

- a phase of 0 if $f<F_c$
- a phase of 180° if $f>F_c$
- an infinite gain (resonance) when $f=F_c$

B.3   High-pass filter cutoff to give flat gain at nominal frequency, when combined with 2 cascaded low-pass anti-aliasing filters

By equations (3.3) & {B.6}, the gain of the total filter package (2 cascaded low-pass filters

plus high-pass filter) will be:

$$Gain = \frac{1}{\left(1 + \left(\dfrac{f}{F_{cLPF}}\right)^2\right)\sqrt{1 + \left(\dfrac{F_{cHPF}}{f}\right)^2}}$$

{B.8}

where F_{cLPF} and F_{cHPF} are the cut-off frequencies for the low-pass and high-pass filters.

Equation {B.8} can be differentiated and solved to find the high-pass filter cut-off frequency required to achieve flat gain at nominal frequency (f), for a given low-pass filter.

$$Gain = \frac{1}{\sqrt{K}} \quad where \quad K = \left(1 + \left(\frac{f}{F_{cLPF}}\right)^2\right)^2 \left(1 + \left(\frac{F_{cHPF}}{f}\right)^2\right)$$

$$\Rightarrow \frac{dGain}{df} = -\frac{1}{2} \cdot \frac{1}{K^{\frac{3}{2}}} \cdot \frac{dK}{df}$$

Now, solve to find F_{cHPF} which makes $\dfrac{dGain}{df} = 0$. This can be done by solving for $\dfrac{dK}{df} = 0$

$$\frac{dK}{df} = 2 \cdot \left(1 + \left(\frac{f}{F_{cLPF}}\right)^2\right) \cdot \frac{2f}{F_{cLPF}^2} \left(1 + \left(\frac{F_{cHPF}}{f}\right)^2\right) + \left(1 + \left(\frac{f}{F_{cLPF}}\right)^2\right)^2 \cdot \frac{-2F_{cHPF}^2}{f^3}$$

$$0 = \frac{2f}{F_{cLPF}^2}\left(1 + \left(\frac{F_{cHPF}}{f}\right)^2\right) + \left(1 + \left(\frac{f}{F_{cLPF}}\right)^2\right) \cdot \frac{-F_{cHPF}^2}{f^3}$$

$$0 = \frac{2f^4}{1}\left(1 + \left(\frac{F_{cHPF}}{f}\right)^2\right) + \left(F_{cLPF}^2 + f^2\right) \cdot \frac{-F_{cHPF}^2}{1}$$

$$0 = F_{cHPF}^2\left(2f^2 - F_{cLPF}^2 - f^2\right) + 2f^4 = F_{cHPF}^2\left(f^2 - F_{cLPF}^2\right) + 2f^4$$

$$F_{cHPF} = \sqrt{\frac{2f^4}{\left(F_{cLPF}^2 - f^2\right)}}$$

Which finally reduces to

$$F_{cHPF} = \frac{\sqrt{2}.f}{\sqrt{\dfrac{F_{cLPF}^2}{f^2} - 1}}$$

{B.9}

B.4 Peak impulse response of two cascaded averaging filters

For two digital averaging filters cascaded, the peak and shape of the impulse response to an input signal one sample long at magnitude 1 can be found via the following thought experiment (which is a shortcut to carrying out the convolution integral):

- The two cascaded filters average the signal for N_1 and N_2 samples respectively.

- When the impulse arrives, the output from the first filter becomes $1/N_1$ and will stay at this level for N_1 frames

- This signal immediately enters the second filter and this begins to rise at a rate of $1/N_2$ times the input signal level, which is $1/N_1$. The rise rate is thus $1/(N_1N_2)$ per frame.

- The rising slope of the output from the second filter will last for the shorter of N_1 or N_2 frames. When this time is passed, the output will plateau for $(N_1+N_2)-2*\min(N_1,N_2)$ frames, before falling in a symmetric fashion to 0, (N_1+N_2) frames after the impulse is applied.

- The plateau height will be $1/(N_1N_2)*\min(N_1,N_2) = 1/\max(N_1,N_2)$

The total "area" of the impulse response is thus $2*\frac{1}{2}*\min(N_1,N_2)*1/\max(N_1,N_2)+ ((N_1+N_2)-2*\min(N_1,N_2))*1/\max(N_1,N_2)$

$= 1/\max(N_1,N_2)*[\min(N_1,N_2)+((N_1+N_2)-2*\min(N_1,N_2))]$

$= 1/\max(N_1,N_2)*[(N_1+N_2)-\min(N_1,N_2)]$

$= 1/\max(N_1,N_2)*[\max(N_1,N_2)] = 1$
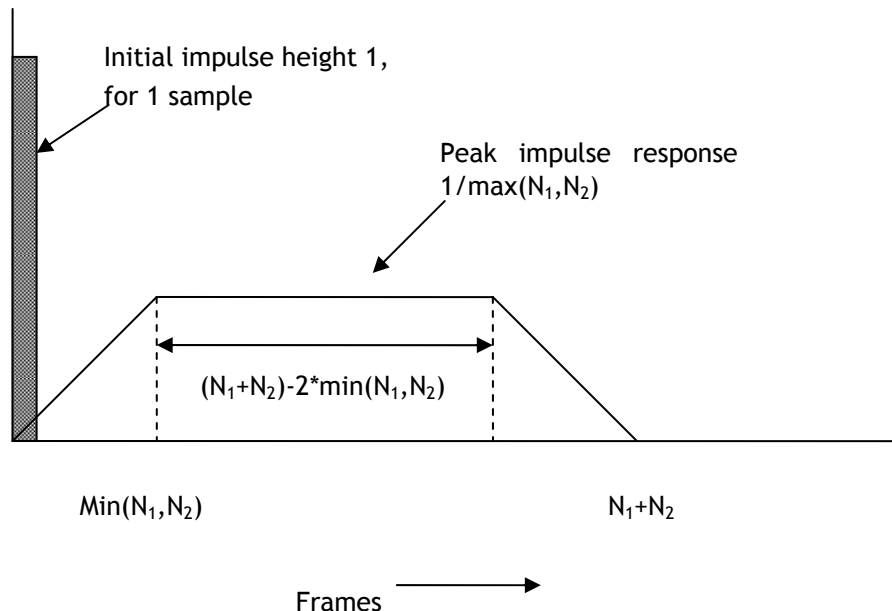
Appendix C   Useful data

C.1   Typical parameters of overhead lines and underground cables

	System voltage	Phase conductor (Al/Fe), mm^2	Resistance Ω/km (20˚C)	Reactance Ω/km (50Hz)	Capacity
Overhead line	400V	25/0	1.06	0.3	150A
Overhead line	400V	50/0	0.64	0.28	250A
Overhead cable	400V	35/0	0.87	0.1	150A
Underground cable	400V	120/0	0.25	0.07	300A
Overhead line	11kV	50/0	0.64	~0.04 *	250A
Underground cable	11kV	185/0	0.16	0.08	380A
Overhead line	20kV	54/9	0.54	~ 0.4 *	250A
Underground cable	20kV	120/0	0.25	0.11	300A
Overhead line	110kV	242/39	0.12	~ 0.4*	650

* - value depends on spacing and cross-arm construction

Tab. C-1 : Typical parameters of overhead lines and underground cables

This data is sourced from the three sources:-

IEE (2003). Electricity distribution network design. IEE Power engineering series 21. 2nd edition. 2003. ISBN 0 86341 309 9. p28.

Wildi, T (2002). Electrical machines, drives and power systems. Fifth edition. Prentice Hall. ISBN 0 13 098637 2. p677.

Laughton, M. & Warne, D. (2003). Electrical engineers reference book. Sixteenth edition. Newnes. ISBN 0 7506 46373. p31/32.

C.2   ITI CBEMA Curve
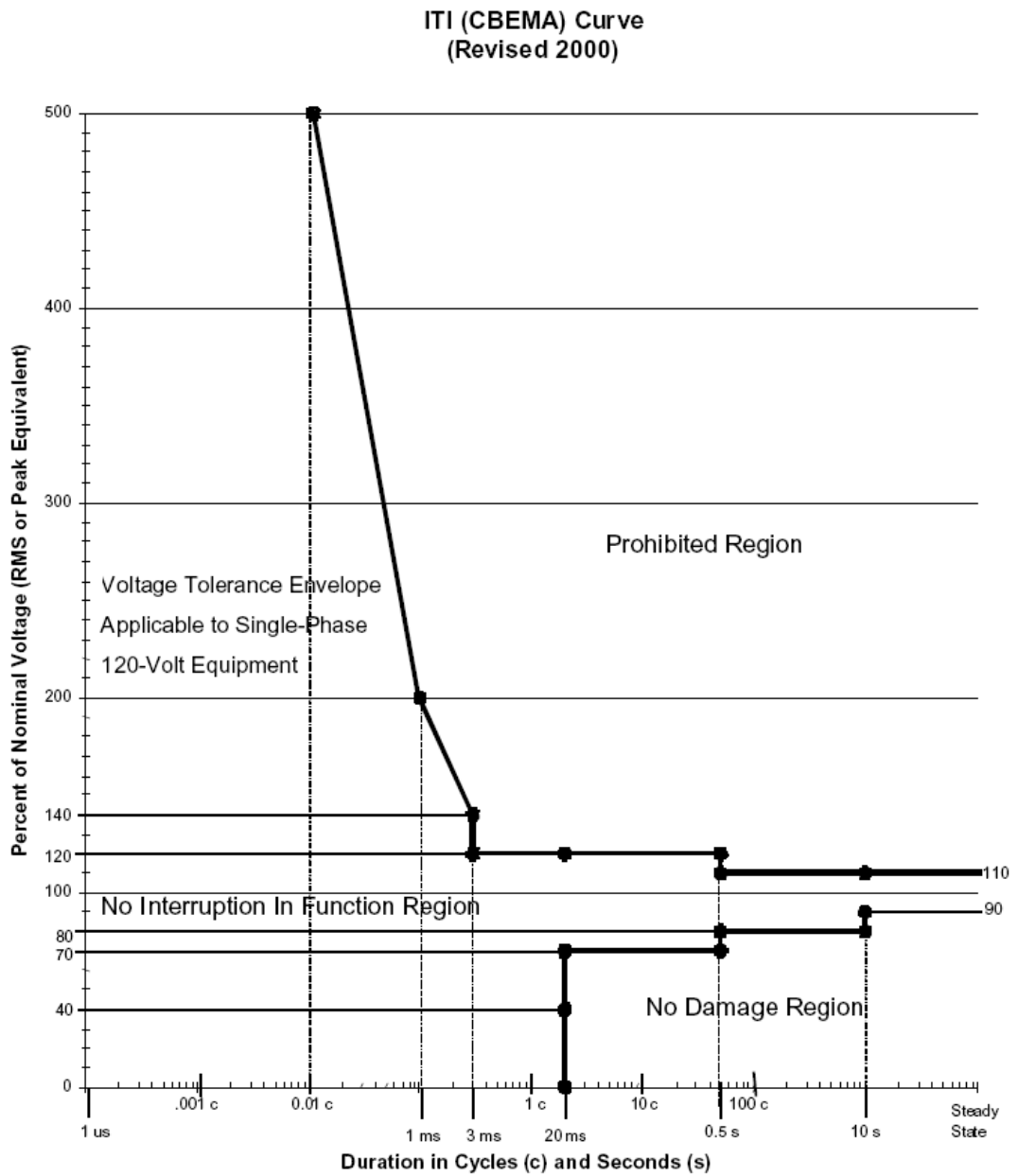
Fig. C-1 : The ITI (CBEMA) Curve

ITI. (2007). ITI (CBEMA) curve application note. [Online]. [Accessed 28/11/2007].
<http://www.itic.org/archives/iticurv.pdf>

Appendix D   Pole-zero & Bode plots of various FIR averaging filter combinations
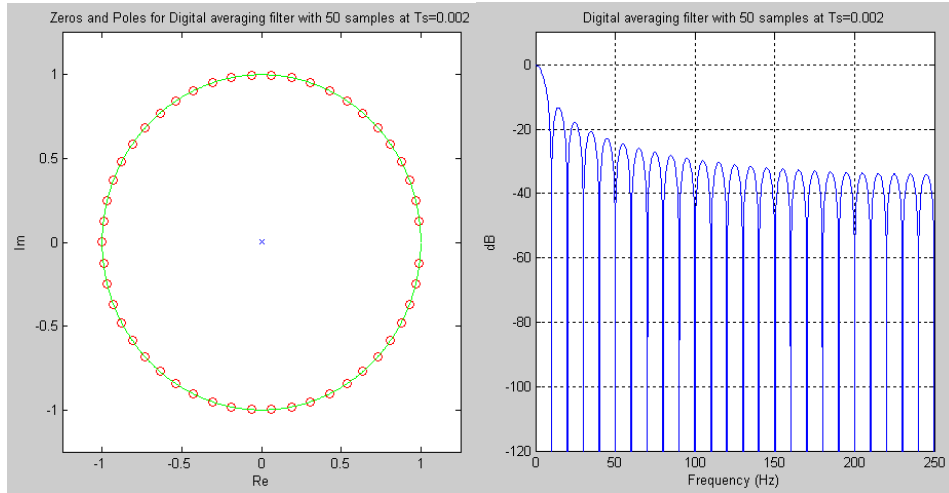
Fig. D-1 : 5 cycle averaging
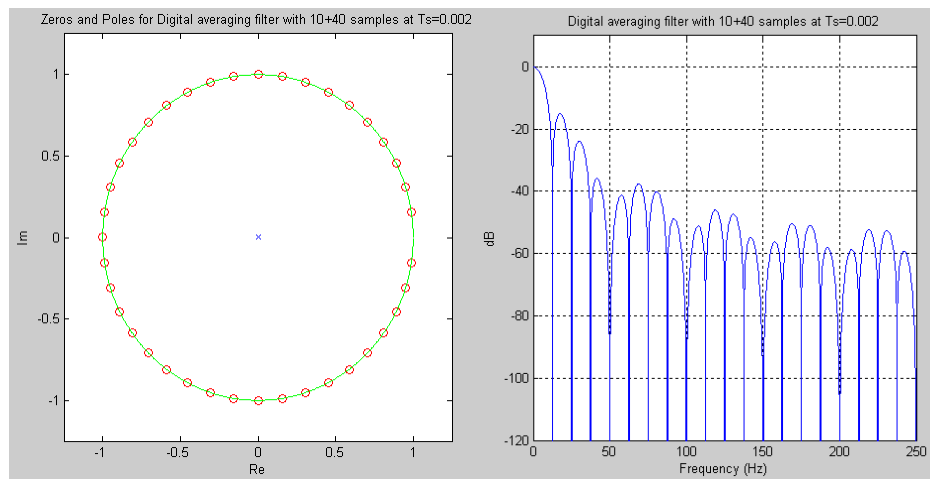
Fig. D-2 : 1+4 cycle averaging
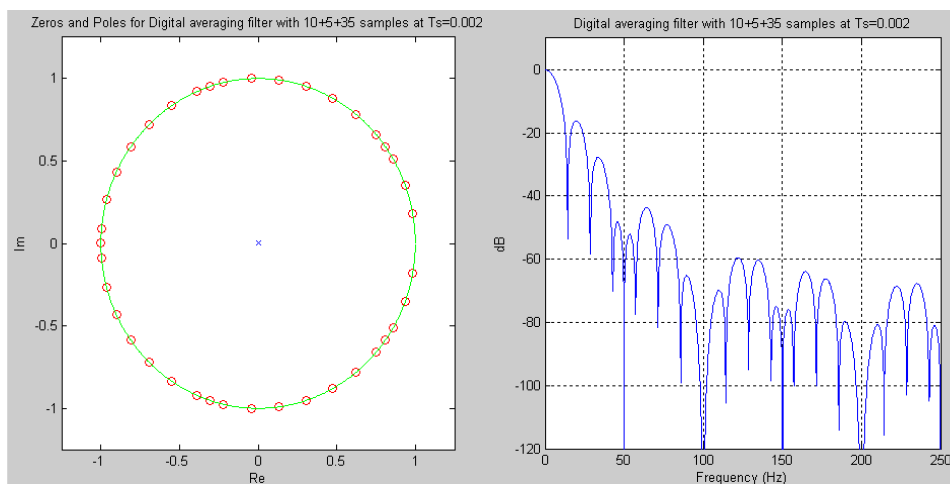
Fig. D-3 : 1+0.5+3.5 cycle averaging
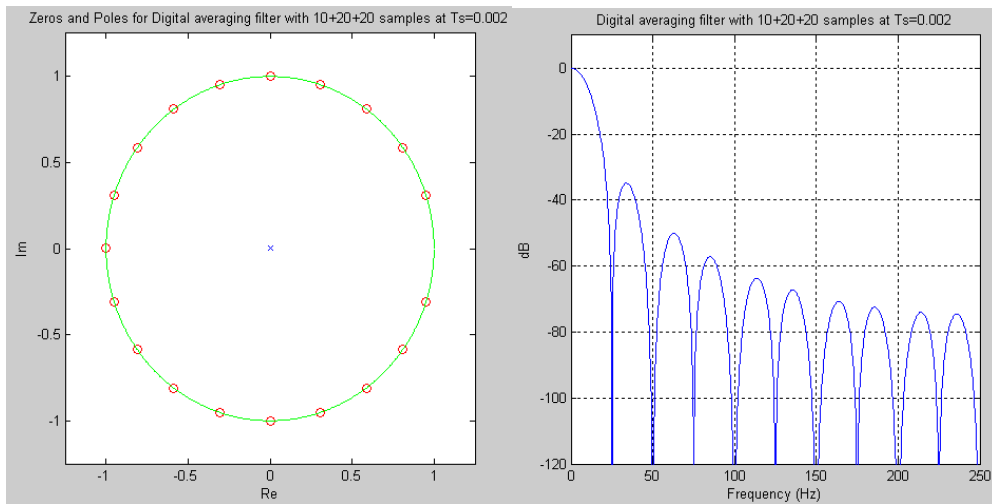
Fig. D-4 : 1+2+2 cycle averaging
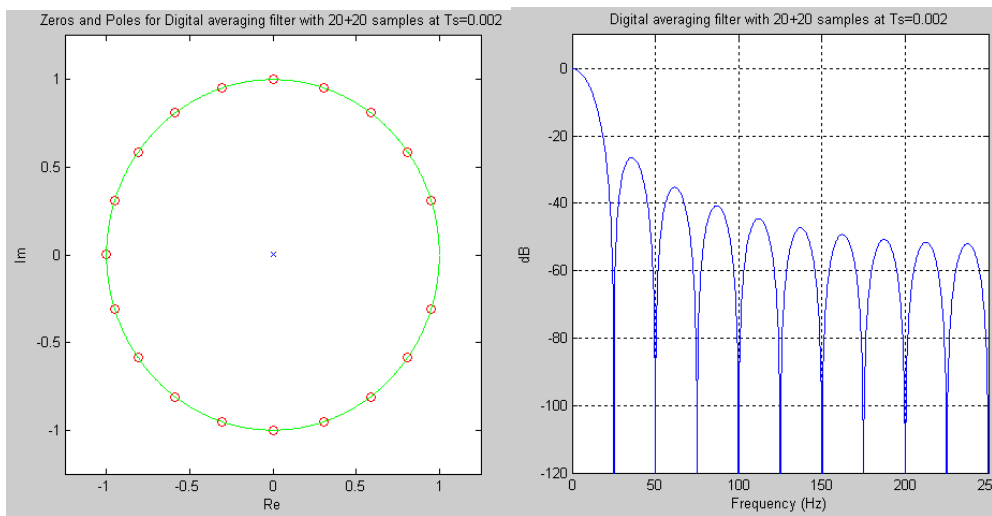
Fig. D-5 : 2+2 cycle averaging

354

Appendix E Archive of code for less successful measurement techniques

E.1 Fixed-reference Fourier frequency measurement with seeding

Three-phase measurement of frequency using three fixed-reference Fourier transforms

Andrew Roscoe, 2007

356
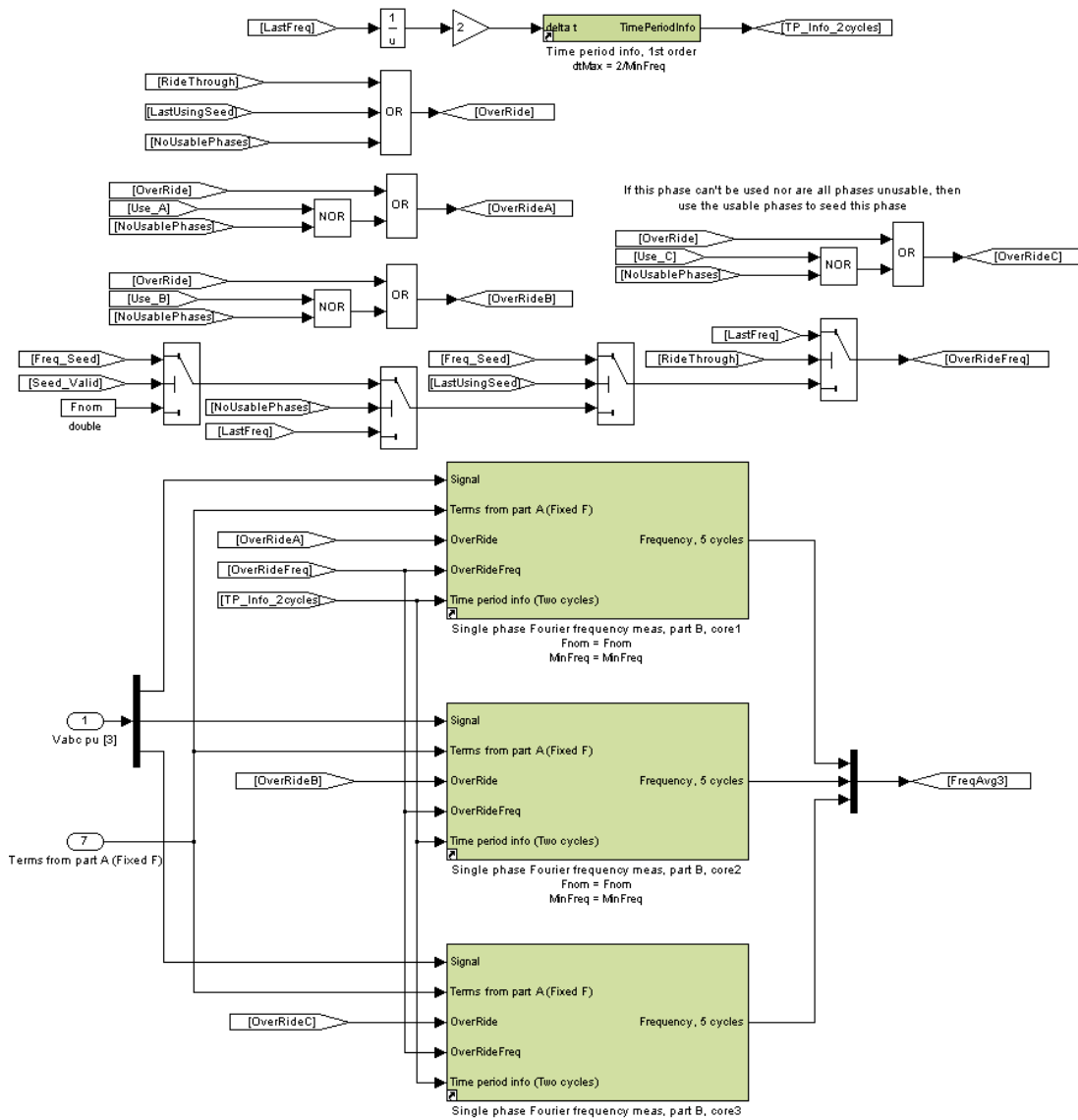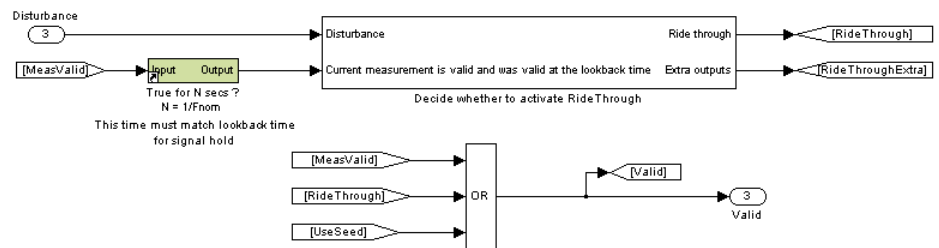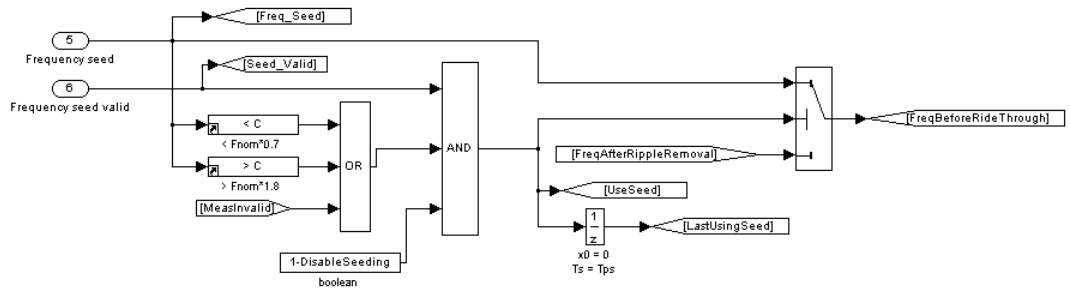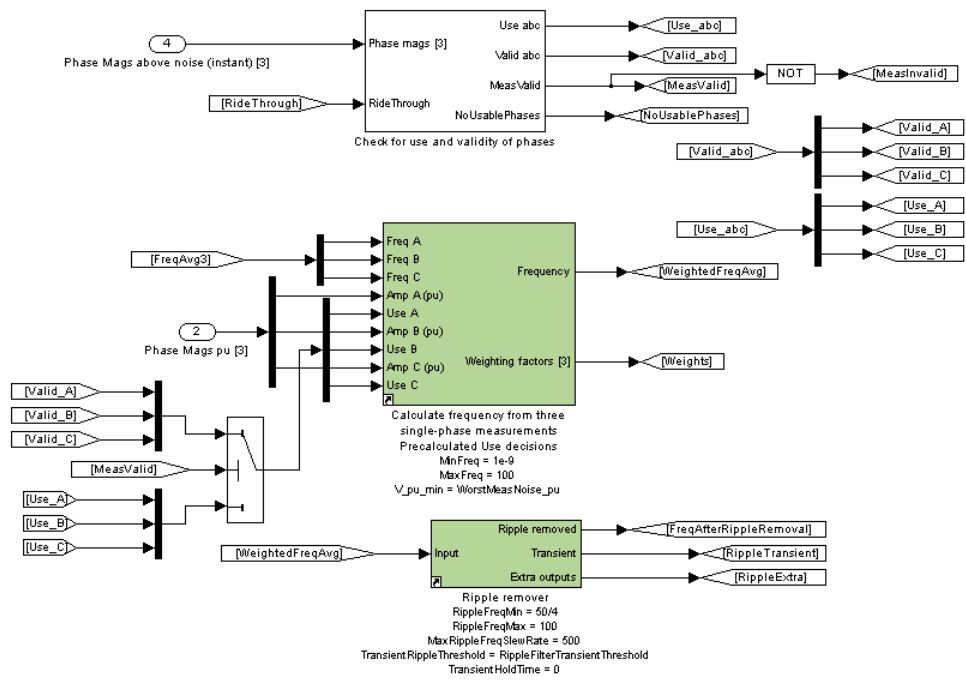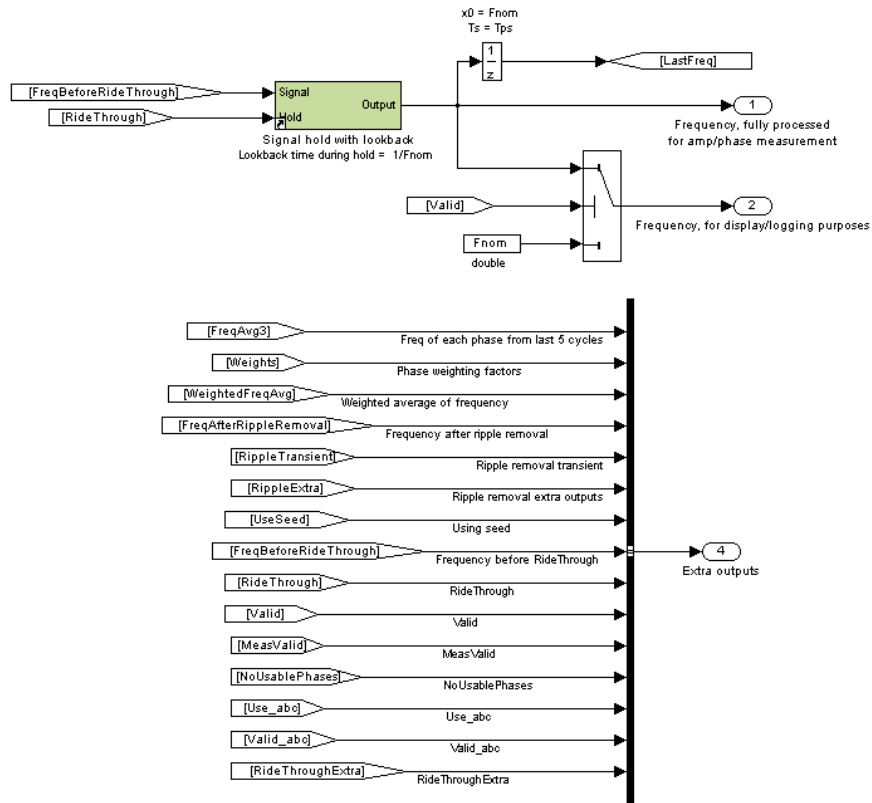
Fig. E-1 : Fixed-reference Fourier frequency measurement – detail (1) spread over this and previous 2 pages
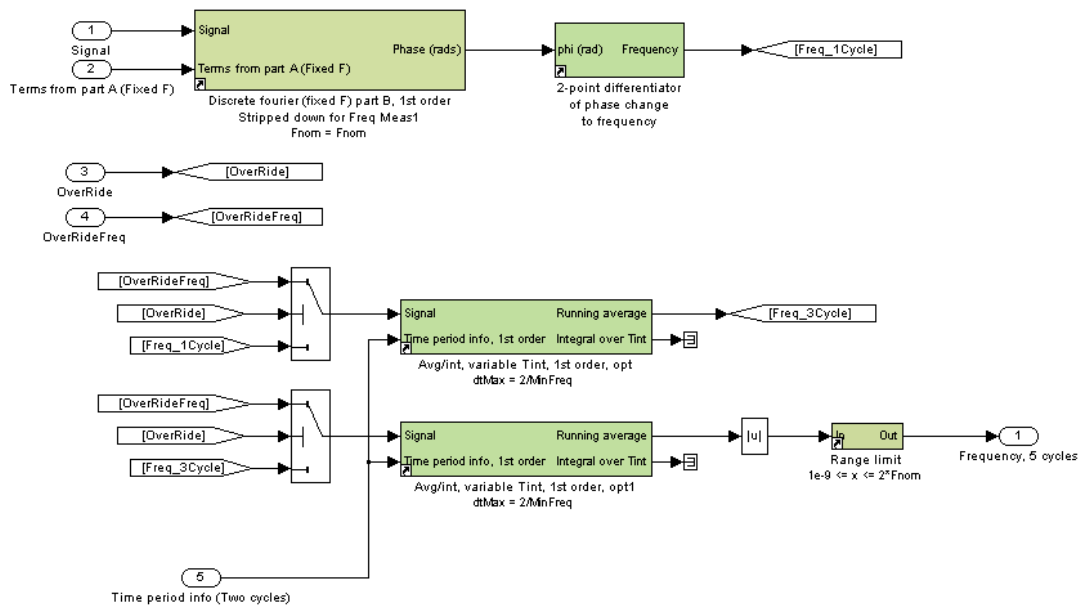
Fig. E-2 : Fixed-reference Fourier frequency measurement – detail (2); core

357

E.2   Three customised single-phase PLLs with weighted averaging and seeding

Three-phase measurement of frequency using three single-phase PLLs

Andrew Roscoe, 2007

Frequency seed

5

[Freq_Seed]

[Seed_Valid]

Frequency seed valid

6

< 35
< Fnom*0.7

> 90
> Fnom*1.8

[PLLValidAndLocked]   NOT

OR

AND

[FreqAfterRippleRemoval]

[UseSeed]

1-DisableSeeding
boolean

[FreqBeforeRideThrough]

Disturbance

3

[Valid]

Input   Output

True for N secs ?
N = 1/Fnom

This time must match lookback time
for signal hold

Disturbance   Ride through

Current measurement is valid and was valid at the lookback time   Extra outputs

Decide whether to activate RideThrough

[RideThrough]

[RideThroughExtra]

[PLLMeasValid]

[UseSeed]
[RideThrough]

[PLLValidAndLocked]

OR

OR

OR

[Output]

3
Valid

[Valid]

x0 = Fnom
Ts = Tps

[FreqBeforeRideThrough]

[RideThrough]

Signal

Hold

Output

Signal hold with lookback
Lookback time during hold = 1/Fnom

In   Out

Range limit
1e-9 <= x <= Fnom*2

1
z

[LastFreq]

1
Frequency, fully processed
for amp/phase measurement

[Output]

Fnom
double

2
Frequency, for display/logging purposes

[ErrAvg_A]

[ErrAvg_B]

[ErrAvg_C]

[FreqAvg3]   Freq of each PLL

ErrAvg abc

[Weights]   Phase weighting factors

[WeightedFreqAvg]   Weighted average of frequency

[WeightedFreqAvgPlus4]   Weighted average of frequency plus 4 cycles averaging

[FreqAfterRippleRemoval]   Frequency after ripple removal

[RippleTransient]   Ripple removal transient

[RippleExtra]   Ripple removal extra outputs

[Using_seed_A]

[Using_seed_B]

[Using_seed_C]

Using seed in PLLs abc

[Lock_abc]   PLL Lock abc

[UseSeed]   Use seed

[FreqBeforeRideThrough]   Frequency before RideThrough

[RideThrough]   RideThrough

[Valid]   Valid

[NoUsablePhases]   NoUsablePhases

[Use_abc]   Use_abc

[Valid_abc]   Valid_abc

[RideThroughExtra]   RideThroughExtra

[TP_Info_1cycle]   TimePeriodInfo_1cycle

[TP_Info_2cycles]   TimePeriodInfo_2cycles

4
Extra outputs

**Fig. E-3 : Three single-phase PLLs with weighted averaging and seeding – detail (1)
spread over this and previous 2 pages**

1-phase PLL, 1 cycle measurement, with external frequency seed for large frequency steps

Note: sin(a) * cos(b)= 1/2 * sin(a+b) + 1/2 * sin(a-b)

Andrew Roscoe, 2007

The Frequency seed should come from a fast responding, robust frequency measurement algorithm such as Clarkes.
The seed validity should be low when appropriate (e.g. during two-phase faults if Clarkes is used as the seed)
To use without seed, set valid LOW at all times.

[OverRide]

0
double

[OverRide]

0
double

1
V(pu)

×

[cos]

Signal*cos

Signal   Running average

Time period info, 1st order   Integral over Tint

Avg/int, variable Tint, 1st order, opt
dtMax = 1/MinFreq

7
TimePeriodInfo (One cycle)

6
Correction factor for <>1pu input

×

|u|

< 0.05

< Vq_LockThreshold

Boolean   Latched output

Latched boolean with timed hysteresis for TRUE/FALSE
TrueConfirmTime = Tlock s
FalseConfirmTime = Tunlock s

4
Lock

5
Error Avg

[Reset_or_Hold]

[ResetLevel]

Error [x]   Out [x]

Reset

ResetLevel [x]   Control P,I [2x]

PI Controller
with reset
Kp = Kp
Ki = Ki
Clip lower, upper = Flimits_pu*Fnom

1
PLL frequency

1
—
z
x0 = 0

Frequency   Angle (rad)

Integrate frequency

[PLL_Freq]

cos

[cos]

2
wt

3
Cos wt

The Frequency seed should come from a fast responding frequency measurment device

2
Freq seed

[Freq_Seed]

3
Seed valid

[Seed_Valid]

4
OverRide

[OverRide]

5
OverRideFreq

[OverRideFreq]

[Freq_Seed]   Freq seed

[PLL_Freq]   PLL frequency

[Seed_Valid]   Seed valid

[OverRide]   OverRide

[OverRideFreq]   OverRideFreq

Reset to seed

Reset/Hold

ResetLevel

6
Using seed freq

[Reset_or_Hold]

[ResetLevel]

Frequency loopback and seeding

Fig. E-4 : Three single-phase PLLs – detail (2); core

Determine whether the PLL needs seeding (possible harmonic/subharmonic lock, or far from lock)
Ensure PI controller is loaded with the RideThrough frequency when fault ride-through is active

Determine whether the PLL needs reseeding:

If real frequency (the seed) is F0, and the PLL frequency is F, and we have N cycles averaging for the q error, then:-

Andrew Roscoe, 2007

Over the N cycles, the dq trajectory will move (F0-F).N/F revolutions over the N cycles, (averaging time N/F).
This is ((F0/F) -1).N revolutions.
The absolute value of this must be less than 1/4 for us to sensibly track the frequency.
Hence, limits on F are:-

F > F0/(1+1/4N)
F < F0/(1-1/4N)

In the algorithm above, N is set to 1
F > F0.4/5
F < F0.4/3

If it is not within these limits, then it should be set to F0

[SeedFreq]

1
Freq seed

4/5

1/(1+1/4N)

4/3

1/(1-1/4N)

2
PLL frequency

<

>

OR

3
Seed valid

AND

Input   Output

True for N secs ?
N = 1/Fnom

[OverRide]

NOT

AND

1
Reset to seed

[ResetToSeed]

[OverRide]

[OverRideFreq]

[ResetToSeed]

[SeedFreq]

Reset - HIGHER priority

Reset value(s) [x] - HIGHER priority

Reset - LOWER priority

Reset value(s) [x] - LOWER priority

Reset_Out

ResetLevel_Out [x]

2
Reset/Hold

3
ResetLevel

Prioritised PI/Integrator reset inputs

4
OverRide

[OverRide]

5
OverRideFreq

[OverRideFreq]

Fig. E-5 : Three single-phase PLLs – detail (3); core PLL seeding decision

E.3 Zero crossings

Three-phase measurement of frequency using zero crossings

Andrew Roscoe, 2007
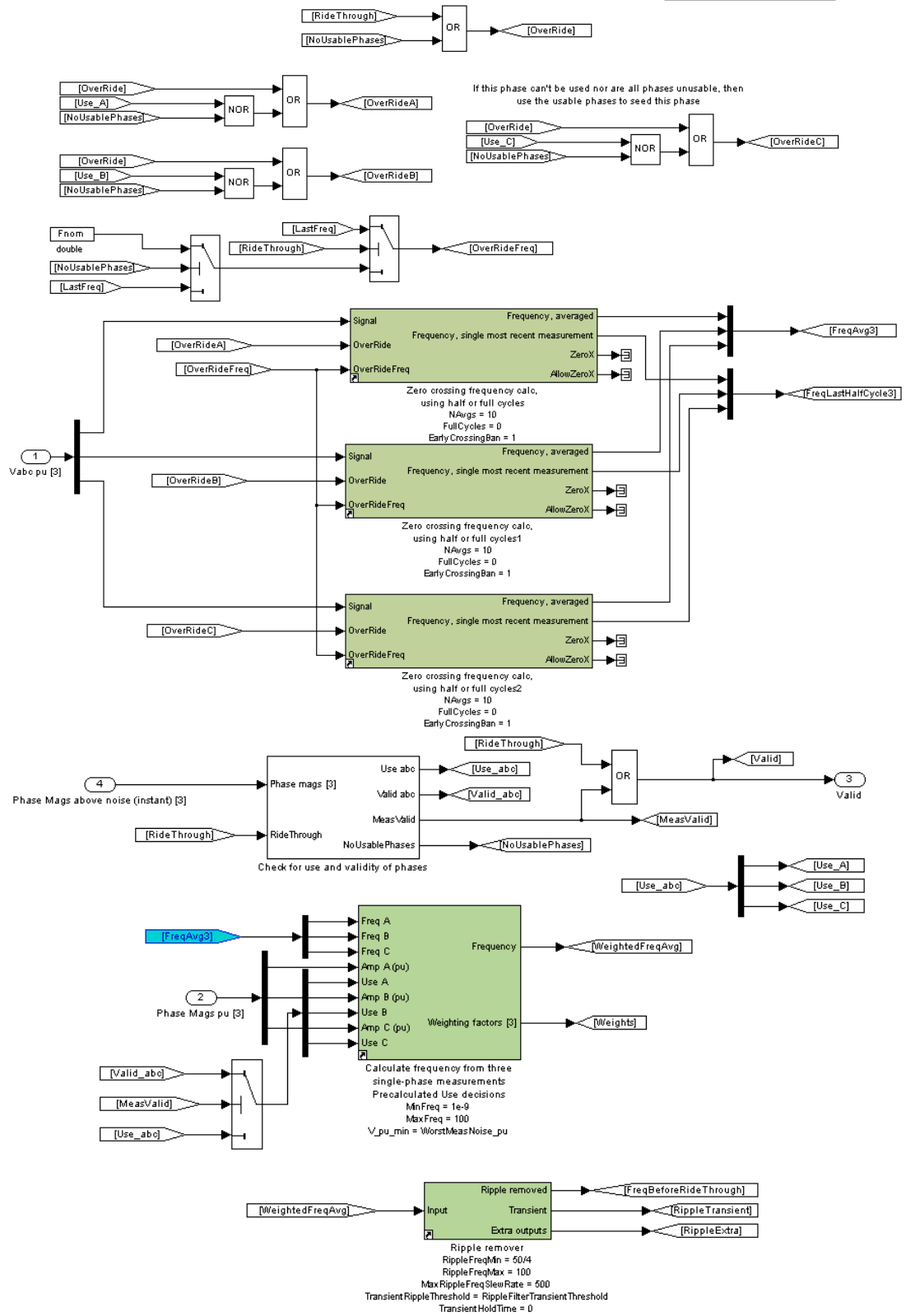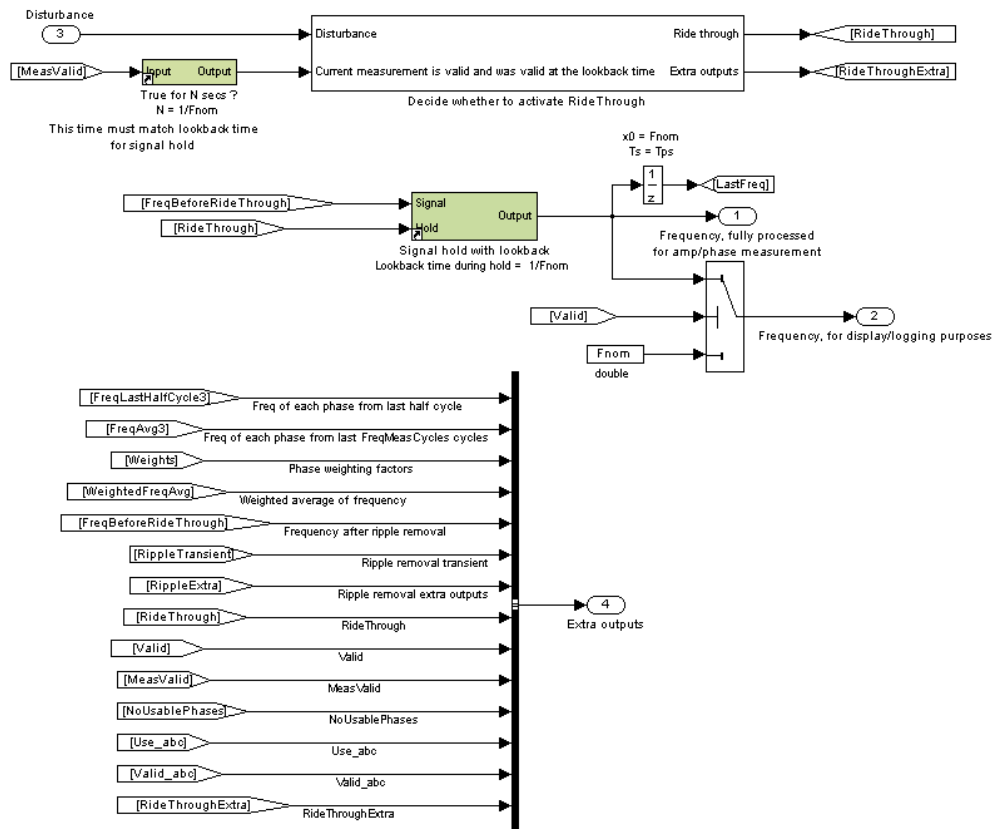
Fig. E-6 : Zero crossing measurement with weighted averaging – detail (1) spread over this and previous page

Frequency measurement by zero crossings over half-cycles or full-cycles

Andrew Roscoe, 2007
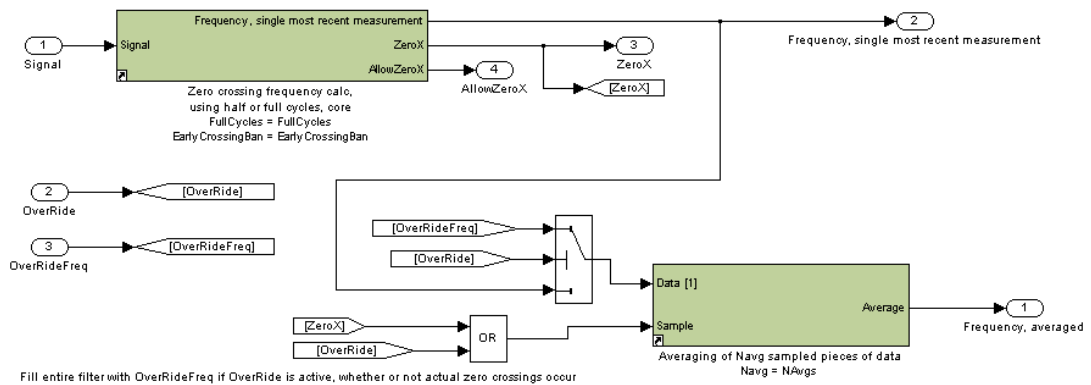
Fig. E-7 : Zero crossing measurement with weighted averaging – detail (2); core

Frequency measurement by zero crossings over half-cycles or full-cycles

Measures the frequency of a signal by using zero crossings (with interpolation).
This new routine will work with low precision arithmetic and for long periods of time
(It is not dependent upon comparisons of system clock outputs)

V1, T1 is the voltage and time just before the zero crossing
V2, T2 is the voltage and time just after the zero crossing

The time estimate for the actual zero crossing is T1 - V1(T2-T1)/(V2-V1)

Hard math function count:-
```
sin/cos        : 0
atan2          : 0
sqrt           : 0
delay buffers  : 0
```

Must make this <0.5 or less so that we don't get stuck on a subharmonic
0.4 is plenty enough for half-cycle measurement, since it will stop all subsequent spurious around the
current zero crossing, until the wave is up at an amplitude of about sin(0.4*180)=sin(72)=0.95
Also allow for 2*Tps less, to allow for discretisation of the sample times

Fig. E-8 : Zero crossing measurement – detail (3); inner core

Appendix F   Examination of the use of sine/cosine lookup tables

To decrease the CPU loading of Fourier calculations, one potential option is to use lookup tables for the evaluations of sine and cosine. This removes the need for the complex mathematical function, at the expense of fast-access RAM use, plus the code required to access the lookup table and interpolate.

To assess the viability of this option, two experiments were carried out:-

1. An analysis of the errors introduced to the 1-cycle and "One plus half" cycle Fourier routines from sections 3.6 and 3.9, by using lookup tables of different sizes.

2. An analysis of the times required for the sine/cosine operations, versus the lookup table options, using the Infineon TC1796 microcontroller.

F.1   Errors introduced due to Sin/Cos lookup tables

The lookup tables in this error analysis are implemented as single-quadrant tables. For example, if the lookup table size is 46, then the lookup table contains 46 pre-calculated values of sin(φ) where φ is [0,2,4,6 … 86,88,90] degrees, with 90/(46-1)=2 degree steps. Results for both sine and cosine evaluations over the full 4 quadrants can be pulled from this table by careful coding, which includes linear interpolation between the most appropriate tabulated values.

The experiments of sections 3.7 and 3.9 were repeated, with the modification that lookup tables of various sizes were used, while sample rate was held at 10 samples per cycle. Only the 1st order Fourier algorithms were analysed.

The conclusion is that the magnitude outputs of the single-cycle Fourier calculation have a mostly DC error term added due to the use of the lookup tables. This can be seen because the ripple frequencies shown in Fig. F-2 are not multiples of the input frequency, but are 0. The effect this has on the overall RMS error magnitudes for the 1-cycle and "One plus Half" cycle measurements are shown in Fig. F-1 and Fig. F-3 . The proportionate effect on the 1-cycle measurements becomes very small once the lookup table size is more than 20 (the lower limit of error is limited by the interpolation/integration error at 10 samples/cycle which is shown in Figure 3-45), but note that this error cannot be removed by the extra half-cycle averaging since it is mostly a DC error term. Thus, to determine an appropriate table size, Fig. F-3 must also be examined. This suggests that a size of 46 (2

degrees per step) reduces the DC error term due to the lookup table to less than 0.0002 pu which is perfectly acceptable. A table of size 21 (4.5 degrees per step) would result in an error of only 0.001pu which is just about acceptable. The maximum errors when evaluating $\sin(\varphi)$ and $\cos(\varphi)$ using an interpolated table with steps of $4.5°$ are a maximum of $8e^{-4}$. If a non-interpolated table was used, the required step spacing to achieve the same magnitude of error can be calculated by knowing that $\sin(\varphi) \approx \varphi$ when φ is small. Hence, the step spacing could be double the maximum error of $8e^{-4}$, i.e. $1.6e^{-3}$ radians, or $0.1°$. By coincidence, this same step spacing would be the requirement for accuracy when calculating the positive and negative sequence values, to keep unbalance measurements accurate to about 0.1%. This is because a set of 3 phase voltages with identical magnitudes, but relative phases of $0°$, $-120.1°$ & $-239.9°$ (i.e. with $0.1°$ phase errors – real or measured), results on a calculated unbalance of 0.1%.
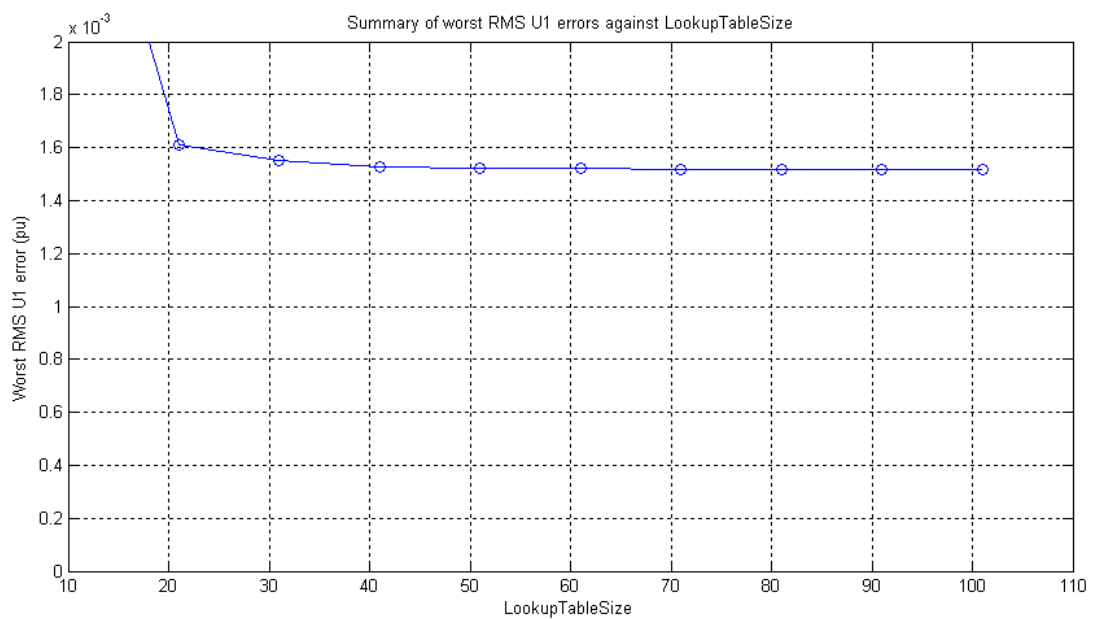
Fig. F-1 : Fourier analysis of fundamental. Largest RMS errors due to Sin/Cos lookup plus integration & interpolation over the 45-55Hz range. Single cycle Fourier analysis.10 samples/cycle, 1st order method.
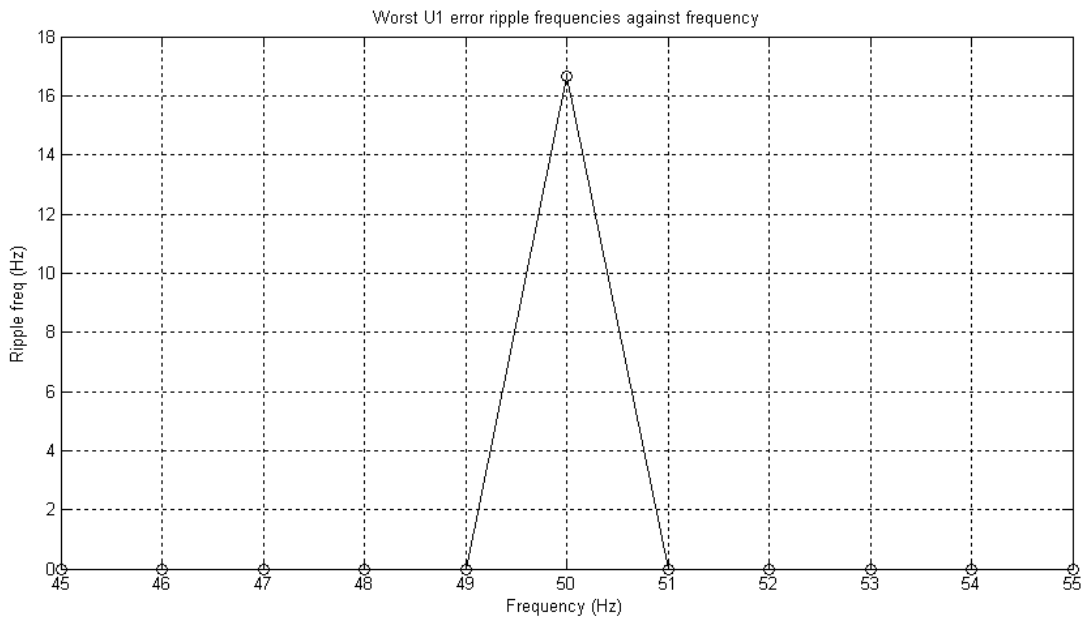
Fig. F-2 : Fourier analysis of fundamental. Ripple frequencies due to Sin/Cos lookup plus integration & interpolation. Lookup table size 11 (9 degree steps). Single cycle Fourier analysis.10 samples/cycle, 1st order method.
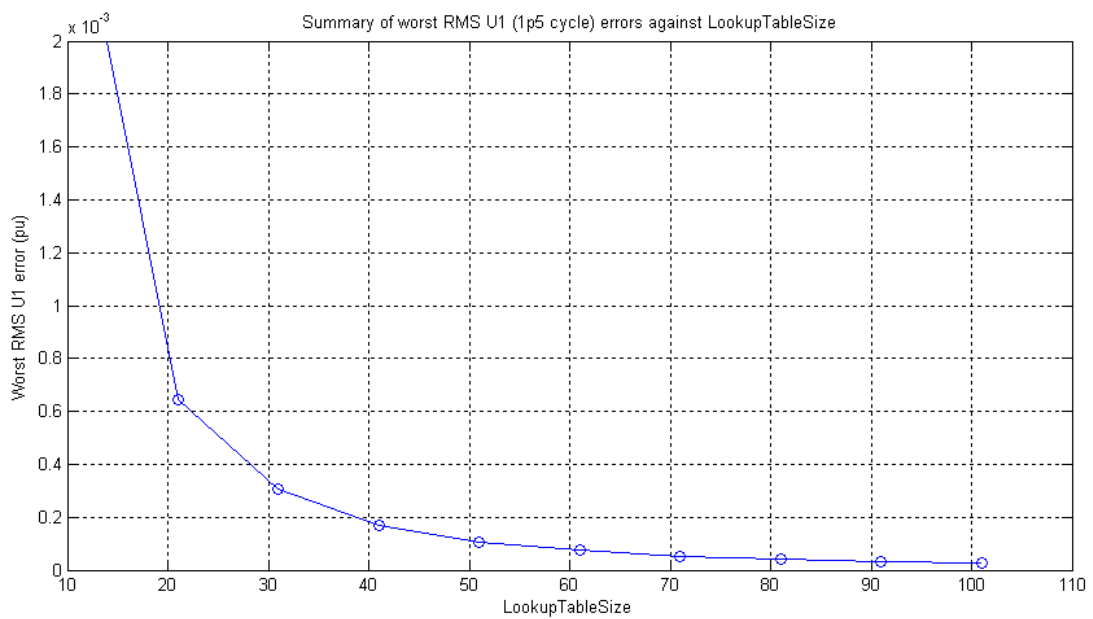
Fig. F-3 : Fourier analysis of fundamental. Largest RMS errors due to Sin/Cos lookup plus integration & interpolation over the 45-55Hz range. Single cycle Fourier analysis plus half cycle averaging.10 samples/cycle, 1st order method.

F.2 Analysis of relative execution times of Sin/Cos vs lookup tables on Infineon TC1796 microcontroller

This is addressed in section Appendix G.

Appendix G   Speed benchmarking on the TC1796 microcontroller and ADI RTS

In Appendix F, the use of sine/cosine lookup tables was proposed as a potential method to reduce the computation time of the Fourier algorithms. To find out whether the improvement could be realised, and to measure the execution times of other pieces of code, a benchmarking framework was created. The benchmark framework can be applied to two types of target, relevant to the ongoing work at the University of Strathclyde:-

- An Infineon TC1796 microcontroller. The benchmark framework was created by stripping out an existing inverter control application to leave only the bare bones of an interrupt-driven code segment at a 4000Hz repetition rate (250μs frame time). This can be reduced to 1000Hz (1000μs frame time) to test very big blocks. Inside this code segment, different pieces of evaluation code can be placed, and repeated in a loop. The number of loop iterations is set so that the execution time approaches 250μs or 1000μs. This makes the execution time measurable using only a basic oscilloscope, coupled to an output pin which the microcontroller toggles at the beginning and end of the loop execution. A simple reference piece of code is used inside the loop the first time, to back the overhead of the looping code out of the measurement. The pieces of code under test are added to this setup in turn, to measure the incremental execution times.

- An ADI RTS real-time station platform. In this application, the execution times for large blocks can be measured by compiling the code segments on to the target processor (either the ce5100 or ce5500 variety – CPU clock speeds 500MHz and 1GHz respectively). The ADI RTS variable ADI_SUBSYS_CUR_TIME[1] can be used to evaluate the incremental execution time when adding code segments.

All the tests presented use code created in Simulink, and auto-generated into C code using the "real-time workshop" and "embedded coder" MATLAB plug-ins. This usually generates very efficient code, but some exceptions of interest are noted here. The results are tabulated in Fig. G-1 to Fig. G-3.

In the case of the Infineon TC1796 microcontroller, The CPU clock speed was 150MHz. The code was initially loaded and executed directly from the internal flash memory at A0000000, which is the normal configuration for a finished application. This causes the code (particularly lookup tables) to execute slower than if it is executed from the internal scratchpad RAM at D4000000. A boot loader application to load the program from flash to RAM at turn-on would improve execution speed, but this would limit the RAM available for

other tasks such as data logging etc. Also note that a power-systems application (plus variable/buffer space) may be too large to all fit inside internal RAM, even without data logging. The results from the initial TC1796 benchmarking are shown in Fig. G-1 and Fig. G-2.

Points of note are:-

- The sine/cosine evaluation from interpolated lookup tables can provide speed benefits over a floating-point calculations, which take about 1µs each. However, the lookup retrieval and interpolation code itself has a reasonable overhead. Interpolating from a memory-efficient 1-quadrant table (0 to 90 degrees only) requires almost as much time as the floating-point calculations. Interpolating from a 4-quadrant table (4 times as much memory required) takes about 0.5 µs for each, even for a pair of sin/cos answers for the same input angle. This would tend to suggest that memory access from the table may be a limiting factor here. A non-interpolated 4-quadrant table can be used to produce results in about 0.3 µs . Again, even when the data is retrieved in sin/cos pairs, for which most of the code is common, the time taken is almost 0.6us. This again suggests that addressing and memory access from the table is the limiting factor. In this case, because the program is loaded into the TC1796 internal flash memory, the lookup table will also be inside the flash memory. A 7x improvement might be realised by running the algorithm (or just storing the table) in internal RAM. In summary, on the TC1796 it is probably not worth using lookup tables when the application is resident in the flash memory, since the times for floating-point calculations are not much longer than the lookup times, and no numerical errors need to be accounted for

- All the rounding functions such as floor/ceil/round/fix provided by Simulink, when compiled on the TC1796, take surprisingly long to evaluate. Also, the floating-point to int32 conversions without saturation checking take longer than the conversions with saturation checking. This is not intuitive, but occurs because the Simulink without-saturation-checking algorithm uses the "fmod" floating-point modulo function, which takes more time to execute than the bounds checking which the saturation checking uses.

- The speed of the rounding functions can be significantly improved by instead using the native casting provided by a C code expression such as "i=(int)f", where I is an int32 and f is a floating-point variable. This can be achieved by using simple Simulink S functions. However, when using this approach, great care must be taken on three counts. The first is that the possibility of overflows/underflows

must be rigorously avoided or trapped. The second is that different target processors behave in different ways. For example, a PC running Simulink responds to the C code casting "i=(int)f" with a "fix" (towards zero) truncation, while the TC1796 processor responds with a "round" action. Within the code libraries used during this thesis, a flag must be set by the user which determines the extra small code segments which must be added to the C code casting to achieve the required rounding function "ceil", "floor", "round" or "fix". These code sections are different for the different target processor types. The final point is that any rounding function may make a mistake of 1, when the floating-point value is very close to a decision threshold. Subsequent code, such as array indexing and use in the "mod" function, should account for this possibility.

- The Simulink library function "mod" is very slow, since it uses the Simulink function "floor", and also because it involves detailed checks for numerical precision around the rounding decision boundaries. The mod function can easily be replaced manually in Simulink using the function mod(A,B)=A-B*floor(A/B), and by using a C cast to carry out the "floor" operation. This produces much faster C code. If required, a saturation/over-range check can be added to make sure that A/B is not too large, before carrying out the "floor" operation. A final tweak is an extra piece of code which checks whether the output of the mod function is actually within the bounds of 0<=mod<=B. This can occur due to numerical precision errors in the "floor" calculation. If the bounds are exceeded, then B can be added or subtracted from the answer to wrap it back inside the expected range.

- The int32 additions/subtractions are very fast, almost immeasurable, as would be expected. Note, however, that the Simulink increment/decrement library functions "V--", "Q--", "V++" & "Q++" did not translate into operations such as "i++" or "i--" in C code. Real-time-workshop instead turned these into "i+1" and "i-1" operations, looking identical to a "bias" of +1 or -1.

- The Simulink algorithms for gain and multiplications using int32 variables are very slow. It is assumed that this accounts for over-ranging etc. If over-ranging is avoided or trapped some other way, or made impossible, then simple single-line C code S-functions of the form k=i*j and k=i/j can be used to carry out multiplications and divisions of int32 variables. These are much faster than the Simulink implementations.

- Floating-point multiplications take no longer than floating-point additions/subtractions (≈0.1 μs), but divisions are 50% longer.

- The "two-taps" variable delay buffer S-function (see section 3.2.1) takes less time

to evaluate (0.25μs) than a floating-point sine or cosine. This is important as many of these blocks are used within the Fourier analysis, averaging, and filtering blocks proposed in this thesis. It should be noted that significant effort has been expended in reducing the execution time of the single-tap and two-tap delay buffers. The final versions take about 0.25μs (almost independent of buffer size), compared to the SimPowerSystems variant which takes 0.75μs (almost independent of buffer size), and the non-S-function method of Figure 3-3 takes massive times of 14.6μs for a 20-delay buffer and 114μs for a 160-delay buffer. This last method uses the "memcpy" instruction in C code which explains the lengthy execution times which are heavily dependent upon buffer length. Similarly to the results of the sin/cos lookup table analysis above, this result suggests that RAM memory access time is one of the major limiting factors of the execution speed on the TC1796, although in this case the memory used for buffers is the internal RAM, which should be 7x faster to access than the internal flash. In addition, the 0.25μs taken for the fastest buffer cannot be explained by simply adding up the execution times for the relatively simple (all int32) operations required for the bulk of the algorithm, so the deduction is that memory access speed is the constraint.

- The "sqrt" function does not take particularly long, at 0.6μs.

- The "atan2" function is relatively expensive, at 1.5μs

- The "abs" function provided by Simulink takes a massive amount of time (0.55μs) and can easily be replaced by Simulink code with a single "switch" and a "unary minus" block, to form an expression y=(a<0?-a:a). This takes only 0.04μs.

- The Simulink function "hypot" contains some obscure c code to avoid over-ranging. Generally, this isn't required if the inputs are reasonable values and a manually coded version is faster/better. hypot=sqrt(a*a+b*b).

- When the output of a Simulink code block is stubbed out with the "terminator" block, the Simulink real-time workshop process which generates C code is extremely efficient at removing all the previous code which is required to calculate that output, if the values are not needed for any other outputs (This feature can be disabled by un-checking the "Block Reduction" optimisation in the Simulink Real-Time-Workshop options). This is very useful to know for future development, since manual effort to cut-down complex blocks to simpler, faster blocks can often be bypassed. Instead, full blocks can be inserted in Simulink and the outputs simply stubbed out, safe in the knowledge that Simulink will cull all the un-needed C code. This can also be used as a "comment" mechanism inside Simulink. A block or section of code with all outputs stubbed with terminators will

generally not appear at all in the C code. Note: an exception is that the S-function delay blocks referred to extensively in this thesis can not be removed in this way. All code up to and including the delay blocks is included inside the c code, whether or not the output of the delay block is actually used. This is because the delay buffers constitute signal storage. Simulink knows there is signal storage within the buffers, and this violates one of the three conditions required for the "Block Reduction" optimisation to be applied. For this reason, some of the large analysis blocks used in this thesis must be manually stripped down if not all functions are required. A good example is that if the all-harmonic RMS and THD measurements are not needed, significant execution time can be saved by manually deleting the un-needed blocks of code in Simulink.

- Another useful observation is that a boolean switch can be inserted inside Simulink code, with the boolean switch value set to a constant. The two paths feeding the switch can be entirely different algorithms. When Simulink creates C code, it knows the value of the constant Boolean flag, and therefore does not create any C code for the un-needed code path. This is very useful, since different candidate algorithms can be switched between at the C code generation stage, without modifying any Simulink code libraries, simply by changing a MATLAB workspace value from a 1 to a 0 or vice versa. As an example, this approach makes the coding of the machine-dependent floating-point to integer casting algorithms easy to manage.

Due to the large number of delay buffer blocks used within some of the algorithms (see section 5.6), significant effort was expended in optimising the delay blocks. Starting from a baseline execution time of 1.5μs per block (the time taken for the SimPowerSystems "variable transport delay" block on the TC1796 without cache enabled), the time for the Author's blocks was reduced to 0.3μs, using the same target configuration. This is achieved by strict "in-lining" of the S-function code within the Simulink ".tlc" file, use of pointer arithmetic, removal of un-needed bounds checks, and careful C-code implementation. The final result is a less readable but faster algorithm with the identical functionality. The code for the two-tap delay output is archived in section G.1. The single-tap (and three-tap) versions are similar.

The execution time of the delay blocks drops to < 0.25μs on the TC1796 when CPU caching is enabled (see below).

Fig. G-1 : Execution times (µs) for common function evaluations on the TC1796 microcontroller. CPU clock 150MHz

Time / us for function execution on TC1796 @ 150Mhz

Categories (top to bottom):
- Rounding functions and conversion to int32 (Simulink)
- Ceil
- Floor
- Round
- Fix (towards zero)
- Ceil -> int32 (no saturation checking)
- Floor -> int32 (no saturation checking)
- Round -> int32 (no saturation checking)
- Fix (towards zero) -> int32 (no saturation checking)
- Ceil -> int32 (with saturation checking)
- Floor -> int32 (with saturation checking)
- Round -> int32 (with saturation checking)
- Fix -> int32 (with saturation checking)
- Simulink MOD (uses Floor) (2.3us)
- Rounding functions and conversion to int32 (manual codings)
- Direct casting to int32 (Ceil)
- Direct casting to int32 (Floor)
- Direct casting to int32 (Round)
- Direct casting to int32 (Round) with saturation check
- Direct casting to int32 (Fix, towards zero rounding)
- Direct casting to Uint32 (Ceil)
- Direct casting to Uint32 (Floor)
- Ceil via casting and reconversion to float
- Floor via casting and reconversion to float
- Round via casting and reconversion to float
- Fix via casting and reconversion to float
- MOD (manual coding with sat and tweak checks)
- MOD (manual coding without sat and tweak checks)
- Int32 operations
- Int32 Reference
- Int32 Increment
- Int32 Decrement
- Int32 Bias
- Int32 Addition
- Int32 Subtraction
- Int32 Gain (simulink)
- Int32 Multiplication (simulink)
- Int32 Multiplication (manual coding)
- Int32 Division (manual coding)

Fig. G-2 : Execution times (µs) for rounding and integer function evaluations on the TC1796 microcontroller. CPU clock 150MHz

A 30% speedup on the TC1796 can be achieved by enabling the CPU cache[1]. This allows

---

[1] To enable the CPU cache on the TC1796, link the application to 0x80000000 not 0xA0000000 in Tasking, and program the flash in HITOP via 0x80000000. Reset and run from 0x80000000. Inside the initialisation C code, insert the lines:-

MAIN_vResetENDINIT();

PMI_CON0=0;/* (Enable 16kB CPU caching) See systems units manual pages 2-24 to 2-27 */

MAIN_vSetENDINIT();

flash memory chunks in the 0xA0000000 flash memory area to be uploaded into the 16kB CPU cache as required, by using the memory mapped section at 0x80000000, to improve execution speed. The improvements affect most of the individual code segments by around 30% relative to the times shown in Fig. G-1 and Fig. G-2.
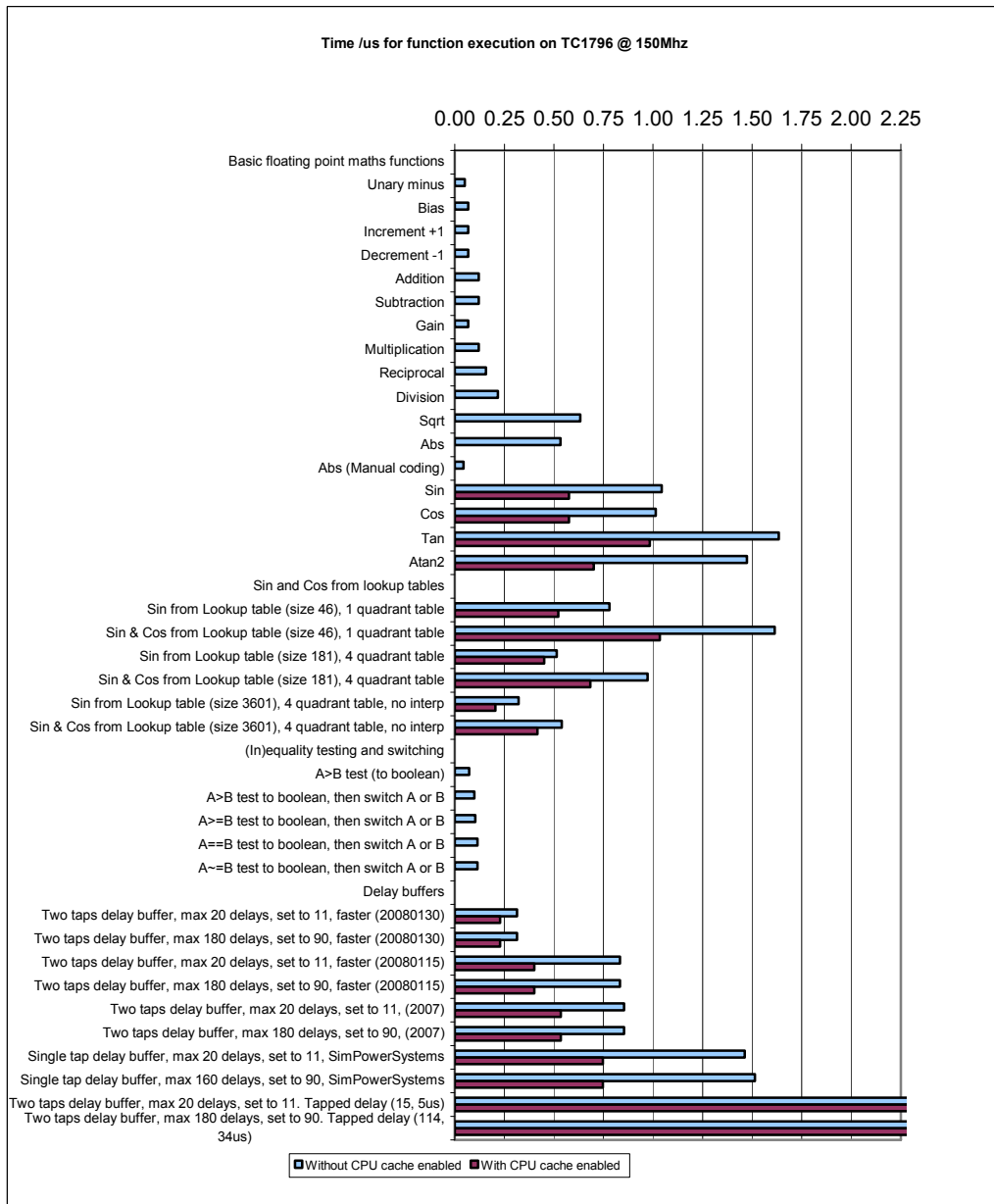
Fig. G-3 : Execution times (µs) for common function evaluations on the TC1796 microcontroller. CPU clock 150MHz. Some functions re-benchmarked with CPU cache enabled.

Fig. G-3 shows the data from Fig. G-2 repeated, again on the TC1796, but with a few of the functions re-benchmarked with the CPU cache enabled for comparison. The most important speed-up is that the delay block execution time can be dropped from 0.3µs to < 0.25µs.

In Fig. G-4, the times for some large algorithms are shown. On this chart, the same algorithms have been benchmarked on three different platforms:-

- Infineon TC1796 microcontroller, CPU clock 150MHz, program in 2MB internal flash memory at 0x80000000, 16kB CPU cache enabled, with the most optimised delay blocks (version 20080201)

- ADI RTS processor, ce5100 variety. Motorola PowerPC, CPU clock 500MHz, with the most optimised delay blocks (version 20080201)

- ADI RTS processor, ce5500 variety, with the most optimised delay blocks (version 20080201). *"CE5500 is ADI's next generation of compute power for the RTS real-time simulator. The heart of this compute engine is the 1GHz G4-based PowerPC processor. This CE includes 32KB on-chip L1 cache, 256KB on-chip L2 cache, 2MB L3 cache, and 512 MB of RAM… Performance improvements range from 3 to 5 times the computational power of the 500MHz CE5100"*

The fourth algorithm with times of 133/323/35 μs contains all the code required to evaluate voltage and power flow at a node of a three-phase power system. Frequency and the voltage magnitudes/phases are calculated as per section 5, with ripple removal filters applied to the 3 voltage magnitude results. The 3-phase current magnitudes and phases are also analysed by re-using the same measured frequency, as per section 3. In addition, a positive/negative sequence analysis is carried out and the unbalance calculated. P & Q flows, power factors and power angles are calculated for the three phases. The sequence analysis and power flow calculations are done without any extra sin/cos evaluations, by careful re-use of path averaging data (see section 3.9).

Finally, Fig. G-5 shows the breakdown of execution time on the TC1796 microcontroller, this time also showing the times required for the DC bias removal, ADC skew, and amplitude/phase corrections for LPF and FIR filter stages. The total times for 3-phase voltage/frequency measurement is 113μs. Extending this to a 6-phase voltage/current set with full sequence/balance analysis and power flow analysis increases the algorithm time to 156μs.

Fig. G-4 : Execution times (µs) for large 3-phase signal processing algorithms on the TC1796 microcontroller, and the ADI RTS processors (ce5100 & ce5500).

Fig. G-5 : Execution time breakdown for nodal voltage/current/power flow analysis algorithm on the TC1796 microcontroller

377

G.1 S-function for the "two-taps" variable-delay buffer

Here, the C-code and "tlc" file for the variable-delay "two-taps" delay buffer is archived. Significant effort has been expended in incremental optimisation and benchmarking of this algorithm, despite its apparent simplicity. Mainly, the effort involves optimising the "tlc" file which is used to generate actual C-code by Real-Time Workshop. The hard part is not the algorithm itself, but optimising the way that Simulink creates the interface between the Simulink code and the "tlc" code. The "tlc" file is fully "in-lined" (see the Simulink documentation), and resulting assembler instructions due to the Simulink-S-Function interface have been minimised as far as possible. The code for the "single-tap" and "three-tap" buffers is very similar, as is the code for a buffer with a fixed delay. Note that in the code shown here, the checks for sensible delay value inputs have been commented out, as the prior Simulink code ensures that the values sent are within acceptable limits. (In the applications of this thesis, the same delay value is sent to many delay blocks, so this check needs to be carried out only once, for many delay blocks). Less robust Simulink algorithms should use S-functions with these checks re-inserted!

G.1.1 SF_VariableDiscreteDelayTwoTapsFaster.c

```
/*
 * File: SF_VariableDiscreteDelayTwoTapsFaster.c
 *
 * Description:
 *   S-function "SF_VariableDiscreteDelayTwoTapsFaster.c".
 *
 *   Author : Andrew Roscoe, 2006-2008
 *            University of Strathclyde
 *
 * This version is the most in-lined, has direct feedthrough
 * (no "normal" S-function state implementation), and also
 * skips the bounds checking of the input parameter "DelaySamples".
 * "DelaySamples" is assumed to already be in the range
 * 1 <= DelaySamples <= MaxDelaySamples
 * If values outside this range are input, segmentation faults may occur.
 *
 * Slower versions of this code can be implemented :-
 * - with less in-lining (clearer, more easily maintainable code)
 * - without direct feedthrough (by adding a S-function state implementation)
 * - with bounds checking of "DelaySamples" re-enabled
 *
 */


#define S_FUNCTION_NAME SF_VariableDiscreteDelayTwoTapsFaster
#define S_FUNCTION_LEVEL 2
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*/
/* %%%-SFUNWIZ_defines_Changes_BEGIN --- EDIT HERE TO _END */
#define NUM_INPUTS          2
/* Input Port  0 */
#define IN_PORT_0_NAME      Signal
#define INPUT_0_WIDTH       1
#define INPUT_DIMS_0_COL    1
#define INPUT_0_DTYPE       real_T
#define INPUT_0_COMPLEX     COMPLEX_NO
#define IN_0_FRAME_BASED    FRAME_NO
#define IN_0_DIMS           1-D
#define INPUT_0_FEEDTHROUGH 1
#define IN_0_ISSIGNED        0
#define IN_0_WORDLENGTH      8
#define IN_0_FIXPOINTSCALING 1
#define IN_0_FRACTIONLENGTH  9
#define IN_0_BIAS            0
```

```
#define IN_0_SLOPE          0.125
/* Input Port  1 */
#define IN_PORT_1_NAME       DelaySamples
#define INPUT_1_WIDTH        1
#define INPUT_DIMS_1_COL     1
#define INPUT_1_DTYPE        int32_T
#define INPUT_1_COMPLEX      COMPLEX_NO
#define IN_1_FRAME_BASED     FRAME_NO
#define IN_1_DIMS            1-D
#define INPUT_1_FEEDTHROUGH 1
#define IN_1_ISSIGNED        0
#define IN_1_WORDLENGTH      8
#define IN_1_FIXPOINTSCALING 1
#define IN_1_FRACTIONLENGTH 9
#define IN_1_BIAS            0
#define IN_1_SLOPE          0.125

#define NUM_OUTPUTS          2
/* Output Port  0 */
#define OUT_PORT_0_NAME       DelayedSignal
#define OUTPUT_0_WIDTH        1
#define OUTPUT_DIMS_0_COL     1
#define OUTPUT_0_DTYPE        real_T
#define OUTPUT_0_COMPLEX      COMPLEX_NO
#define OUT_0_FRAME_BASED     FRAME_NO
#define OUT_0_DIMS            1-D
#define OUT_0_ISSIGNED         1
#define OUT_0_WORDLENGTH       8
#define OUT_0_FIXPOINTSCALING 1
#define OUT_0_FRACTIONLENGTH  3
#define OUT_0_BIAS             0
#define OUT_0_SLOPE           0.125
/* Output Port  1 */
#define OUT_PORT_1_NAME       DelayedSignalTwo
#define OUTPUT_1_WIDTH        1
#define OUTPUT_DIMS_1_COL     1
#define OUTPUT_1_DTYPE        real_T
#define OUTPUT_1_COMPLEX      COMPLEX_NO
#define OUT_1_FRAME_BASED     FRAME_NO
#define OUT_1_DIMS            1-D
#define OUT_1_ISSIGNED         1
#define OUT_1_WORDLENGTH       8
#define OUT_1_FIXPOINTSCALING 1
#define OUT_1_FRACTIONLENGTH  3
#define OUT_1_BIAS             0
#define OUT_1_SLOPE           0.125

#define NPARAMS              1
/* Parameter  1 */
#define PARAMETER_0_NAME      MaxDelaySamples
#define PARAMETER_0_DTYPE     int32_T
#define PARAMETER_0_COMPLEX   COMPLEX_NO

#define SAMPLE_TIME_0        INHERITED_SAMPLE_TIME
#define NUM_DISC_STATES      0
#define DISC_STATES_IC       [0]
#define NUM_CONT_STATES      0
#define CONT_STATES_IC       [0]

#define SFUNWIZ_GENERATE_TLC 1
#define SOURCEFILES "__SFB__"
#define PANELINDEX           6
#define USE_SIMSTRUCT        0
#define SHOW_COMPILE_STEPS   0
#define CREATE_DEBUG_MEXFILE 0
#define SAVE_CODE_ONLY       0
#define SFUNWIZ_REVISION     3.0
/* %%%-SFUNWIZ_defines_Changes_END --- EDIT HERE TO _BEGIN */
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*/
#include "simstruc.h"
#define PARAM_DEF0(S) ssGetSFcnParam(S, 0)

#define IS_PARAM_INT32(pVal) (mxIsNumeric(pVal) && !mxIsLogical(pVal) &&\
!mxIsEmpty(pVal) && !mxIsSparse(pVal) && !mxIsComplex(pVal) && mxIsInt32(pVal))

/*=====================*
 * S-function methods *
```

```
 *=====================*/
#define MDL_CHECK_PARAMETERS
 #if defined(MDL_CHECK_PARAMETERS) && defined(MATLAB_MEX_FILE)
   /* Function: mdlCheckParameters =============================================
     * Abstract:
     *    Validate our parameters to verify they are okay.
     */
    static void mdlCheckParameters(SimStruct *S)
    {
     #define PrmNumPos 46
     int paramIndex = 0;
     bool validParam = false;
     char paramVector[] ={'1'};
     static char parameterErrorMsg[] ="The data type and/or complexity of parameter    does not match the information "
     "specified in the S-function Builder dialog. For non-double parameters you will need to cast them using int8, int16,"
     "int32, uint8, uint16, uint32 or boolean.";

      /* All parameters must match the S-function Builder Dialog */


            {
             const mxArray *pVal0 = ssGetSFcnParam(S,0);
             if (!IS_PARAM_INT32(pVal0)) {
               validParam = true;
               paramIndex = 0;
               goto EXIT_POINT;
             }
            }
     EXIT_POINT:
      if (validParam) {
             parameterErrorMsg[PrmNumPos] = paramVector[paramIndex];
             ssSetErrorStatus(S,parameterErrorMsg);
      }
           return;
    }
 #endif /* MDL_CHECK_PARAMETERS */
/* Function: mdlInitializeSizes ===============================================
 * Abstract:
 *   Setup sizes of the various vectors.
 */
static void mdlInitializeSizes(SimStruct *S)
{
    const int32_T   *MaxDelaySamples  = mxGetData(PARAM_DEF0(S));

    DECL_AND_INIT_DIMSINFO(inputDimsInfo);
    DECL_AND_INIT_DIMSINFO(outputDimsInfo);
    ssSetNumSFcnParams(S, NPARAMS);  /* Number of expected parameters */
      #if defined(MATLAB_MEX_FILE)
            if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {
             mdlCheckParameters(S);
             if (ssGetErrorStatus(S) != NULL) {
               return;
             }
            } else {
             return; /* Parameter mismatch will be reported by Simulink */
            }
      #endif

    ssSetNumContStates(S, NUM_CONT_STATES);
    ssSetNumDiscStates(S, NUM_DISC_STATES);

    if (!ssSetNumInputPorts(S, NUM_INPUTS)) return;
    /*Input Port 0 */
    ssSetInputPortWidth(S,  0, INPUT_0_WIDTH); /* */
    ssSetInputPortDataType(S, 0, SS_DOUBLE);
    ssSetInputPortComplexSignal(S,  0, INPUT_0_COMPLEX);
    ssSetInputPortDirectFeedThrough(S, 0, INPUT_0_FEEDTHROUGH);
    ssSetInputPortRequiredContiguous(S, 0, 1); /*direct input signal access*/

    /*Input Port 1 */
    ssSetInputPortWidth(S,  1, INPUT_1_WIDTH); /* */
    ssSetInputPortDataType(S, 1, SS_INT32);
    ssSetInputPortComplexSignal(S,  1, INPUT_1_COMPLEX);
    ssSetInputPortDirectFeedThrough(S, 1, INPUT_1_FEEDTHROUGH);
    ssSetInputPortRequiredContiguous(S, 1, 1); /*direct input signal access*/
```

```
    if (!ssSetNumOutputPorts(S, NUM_OUTPUTS)) return;
    /* Output Port 0 */
    ssSetOutputPortWidth(S, 0, OUTPUT_0_WIDTH);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);
    ssSetOutputPortComplexSignal(S, 0, OUTPUT_0_COMPLEX);
    /* Output Port 1 */
    ssSetOutputPortWidth(S, 1, OUTPUT_1_WIDTH);
    ssSetOutputPortDataType(S, 1, SS_DOUBLE);
    ssSetOutputPortComplexSignal(S, 1, OUTPUT_1_COMPLEX);

    /**** Work Vector Modifications ***/
    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, *MaxDelaySamples+1);
    ssSetNumIWork(S, 1);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    /* Take care when specifying exception free code - see sfuntmpl_doc.c */
    ssSetOptions(S, (SS_OPTION_EXCEPTION_FREE_CODE |
                     SS_OPTION_USE_TLC_WITH_ACCELERATOR |
                         SS_OPTION_WORKS_WITH_CODE_REUSE));
}


# define MDL_SET_INPUT_PORT_FRAME_DATA
static void mdlSetInputPortFrameData(SimStruct  *S,
                                     int_T       port,
                                     Frame_T    frameData)
{
    ssSetInputPortFrameData(S, port, frameData);
}
/* Function: mdlInitializeSampleTimes =========================================
 * Abstract:
 *    Specifiy  the sample time.
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, SAMPLE_TIME_0);
    ssSetOffsetTime(S, 0, 0.0);
}


#define MDL_START  /* Change to #undef to remove function */
#if defined(MDL_START)
/* Function: mdlStart =======================================================
* Abstract:
*    This function is called once at start of model execution. If you
*    have states that should be initialized once, this is the place
*    to do it.
*/
static void mdlStart(SimStruct *S)
{
    const int32_T  *MaxDelaySamples  = mxGetData(PARAM_DEF0(S));
    int32_T i;

    for (i = 0; i < (*MaxDelaySamples+1); i++)
        ssGetRWork(S)[i] = 0.0;
    ssGetIWork(S)[0] = 0;                                  /* This is the index of the buffer where we want the data to start going
into */

    mexPrintf("Initialised a SF_VariableDiscreteDelayTwoTapsFaster buffer of %d samples.\n", *MaxDelaySamples+1);
}
#endif /*  MDL_START */

#define MDL_SET_INPUT_PORT_DATA_TYPE
static void mdlSetInputPortDataType(SimStruct *S, int port, DTypeId dType)
{
    ssSetInputPortDataType( S, 0, dType);
}
#define MDL_SET_OUTPUT_PORT_DATA_TYPE
static void mdlSetOutputPortDataType(SimStruct *S, int port, DTypeId dType)
{
    ssSetOutputPortDataType(S, 0, dType);
}


#define MDL_SET_DEFAULT_PORT_DATA_TYPES
static void mdlSetDefaultPortDataTypes(SimStruct *S)
{
  ssSetInputPortDataType( S, 0, SS_DOUBLE);
```

```
 ssSetOutputPortDataType(S, 0, SS_DOUBLE);
}
/* Function: mdlOutputs =========================================================
 *
 */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    const real_T   *Signal          = (const real_T*) ssGetInputPortSignal(S,0);
    const int32_T  *DelaySamples     = (const int32_T*) ssGetInputPortSignal(S,1);
    real_T         *DelayedSignal    = (real_T *)ssGetOutputPortRealSignal(S,0);
    real_T         *DelayedSignalTwo = (real_T *)ssGetOutputPortRealSignal(S,1);
    const int32_T  *MaxDelaySamples  = mxGetData(PARAM_DEF0(S));

    real_T  *VDD_buffer = ssGetRWork(S);

/* VDD_in is a pointer to the index of the buffer where we want the new data to go */
/* *VDD_in is the index of the buffer where we want the new data to go */
    int_T   *VDD_in     = ssGetIWork(S);

    int_T   VDD_out;
    real_T *Out_ptr;
/*   int_T   Offset  */

    VDD_buffer[*VDD_in] = *Signal1;

/*   Offset = *DelaySamples;  */
/*   if (Offset > *MaxDelaySamples) Offset = *MaxDelaySamples;    Check omitted in this faster implementation */
/*   if (Offset < 1) Offset = 1;                                  Check omitted in this faster implementation */

    VDD_out = *VDD_in - *DelaySamples;
    if (VDD_out < 0) {
        VDD_out += *MaxDelaySamples;    /* VDD_out=VDD_out+*MaxDelaySamples+1 */
        VDD_out++;
    }

    Out_ptr=&VDD_buffer[VDD_out];
    *DelayedSignal = *Out_ptr;

    if (VDD_out == *MaxDelaySamples)
        *DelayedSignalTwo = VDD_buffer[0];
    else {
        Out_ptr++;
        *DelayedSignalTwo = *Out_ptr;
    }

    /* Leave VDD_in ready to take the next sample */
    (*VDD_in)++;
    if (*VDD_in > *MaxDelaySamples)
        *VDD_in = 0;
}


/* Function: mdlTerminate =========================================================
 * Abstract:
 *    In this function, you should perform any actions that are necessary
 *    at the termination of a simulation.  For example, if memory was
 *    allocated in mdlStart, this is the place to free it.
 */
static void mdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

G.1.2   SF_VariableDiscreteDelayTwoTapsFaster.tlc

```
SF_VariableDiscreteDelayTwoTapsFaster.tlc
%% File : SF_VariableDiscreteDelayTwoTapsFaster.tlc
%%
%% Description:
%%    S-function "SF_VariableDiscreteDelayTwoTapsFaster.c".
```

```
%%
%%   Author : Andrew Roscoe, 2006-2008
%%            University of Strathclyde
%%
%% This version is the most in-lined, has direct feedthrough
%% (no "normal" S-function state implementation), and also
%% skips the bounds checking of the input parameter "DelaySamples".
%% "DelaySamples" is assumed to already be in the range
%% 1 <= DelaySamples <= MaxDelaySamples
%% If values outside this range are input, segmentation faults may occur.
%%
%% Slower versions of this code can be implemented :-
%% - with less in-lining (clearer, more easily maintainable code)
%% - without direct feedthrough (by adding a S-function state implementation)
%% - with bounds checking of "DelaySamples" re-enabled


%implements  SF_VariableDiscreteDelayTwoTapsFaster "C"



%% Function: Start ======================================================
%%
%% Purpose:
%%      Initialise work vectors (global variables).
%%
%function Start(block, system) Output

  /* S-Function Initialise "SF_VariableDiscreteDelayTwoTapsFaster" Block: %<Name> */

  %assign nelements1 = LibBlockParameterSize(P1)
  %assign param_width1 = nelements1[0] * nelements1[1]
  %if (param_width1) > 1
   %assign pp1 = LibBlockMatrixParameterBase(P1)
  %else
   %assign pp1 = LibBlockParameter(P1, "", "", 0)
  %endif

  %assign vdd_buffer_RWork = LibBlockRWork(0, "", "", 0)
  %assign vdd_in_IWork = LibBlockIWork(0, "", "", 0)


  {
%%    int_T      MaxDelaySamples = %<pp1>;               /* MaxDelaySamples parameter value */
    real_T     *VDD_buffer    = &%<vdd_buffer_RWork>; /* VDD_buffer real work vector */
    int_T      *VDD_in        = &%<vdd_in_IWork>;     /* VDD_in integer work vector */
    int_T      i;

    for (i = 0; i < (%<pp1>+1); i++)
        VDD_buffer[i] = 0.0;
    *VDD_in = 0;       /* This is the index of the buffer where we want the data to start going into */
  }


%endfunction %% Start

%% Function: Outputs ==========================================================
%%
%% Purpose:
%%      Code generation rules for mdlOutputs function.
%%
%function Outputs(block, system) Output
  /* S-Function Output "SF_VariableDiscreteDelayTwoTapsFaster" Block: %<Name> */

  %assign nelements1 = LibBlockParameterSize(P1)
  %assign param_width1 = nelements1[0] * nelements1[1]
  %if (param_width1) > 1
     %assign pp1 = LibBlockMatrixParameterBase(P1)
   %else
     %assign pp1 = LibBlockParameter(P1, "", "", 0)
   %endif

  %assign pu0 = LibBlockInputSignal1(0, "", "", 0)
  %assign pu1 = LibBlockInputSignal1(1, "", "", 0)
  %assign py0 = LibBlockOutputSignal1(0, "", "", 0)
  %assign py1 = LibBlockOutputSignal1(1, "", "", 0)

  %assign vdd_buffer_RWork = LibBlockRWork(0, "", "", 0)
  %assign vdd_in_IWork = LibBlockIWork(0, "", "", 0)

  {
```

```
%%    int_T   MaxDelaySamples   = %<pp1>;                /* MaxDelaySamples parameter value */
%%    real_T  Signal            = %<pu0>;                /* Input signal */
%%    int_T   DelaySamples       = %<pu1>;                /* Input signal - number of delay samples */
%%    real_T  DelayedSignal      = %<py0>;                /* Output signal */
%%    real_T  DelayedSignalTwo   = %<py1>;                /* Output signal */
   real_T  *VDD_buffer          = &%<vdd_buffer_RWork>;  /* VDD_buffer real work vector */
   int_T   *VDD_in              = &%<vdd_in_IWork>;      /* VDD_in integer work vector */
   int_T   VDD_out;
   real_T  *Out_ptr;
/*    int_T   Offset;  */


/* DelayBufferCountTag    This comment allows a count of these blocks in the RTW c-code file */


   VDD_buffer[*VDD_in] = %<pu0>;


/*    Offset = %<pu1>;  */
/*    if (Offset > %<pp1>) Offset = %<pp1>;    Check omitted in this faster implementation */
/*    if (Offset < 1) Offset = 1;              Check omitted in this faster implementation */


   VDD_out = *VDD_in - %<pu1>;
   if (VDD_out < 0) {
       VDD_out += %<pp1>;    /* VDD_out=VDD_out+MaxDelaySamples+1 */
       VDD_out++;
   }


   Out_ptr=&VDD_buffer[VDD_out];
   %<py0> = *Out_ptr;

   if (VDD_out == %<pp1>)
       %<py1> = VDD_buffer[0];
   else {
       Out_ptr++;
       %<py1> = *Out_ptr;
   }


   /* Leave VDD_in ready to take the next sample */
   (*VDD_in)++;
   if (*VDD_in > %<pp1>)
       *VDD_in = 0;
 }


%endfunction %% Outputs



%% [EOF] SF_VariableDiscreteDelayTwoTapsFaster.tlc
```

384

Appendix H   Logged domestic voltage and current waveforms

The figures below record a snapshot of single-phase domestic voltage and current waveforms on 21st March 2008 at the Author's home. The load is relatively light (~500W), and is made up mostly of "energy saver" light bulbs and audio-visual equipment. Of note is the extremely high harmonic content (52.9% THD) of the current drawn, which is mostly from the peak of the voltage waveform.
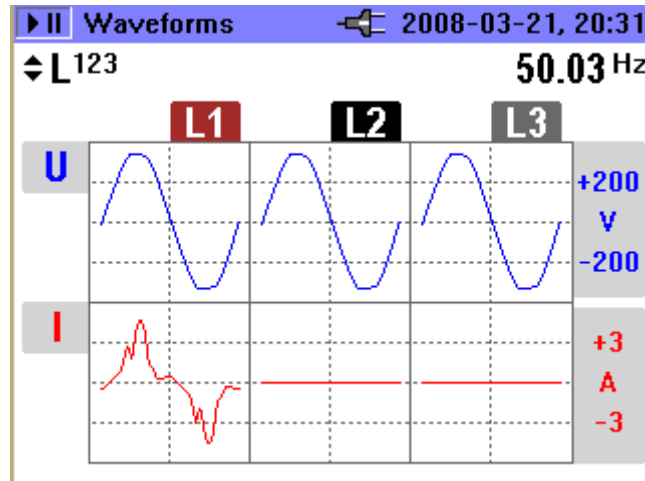

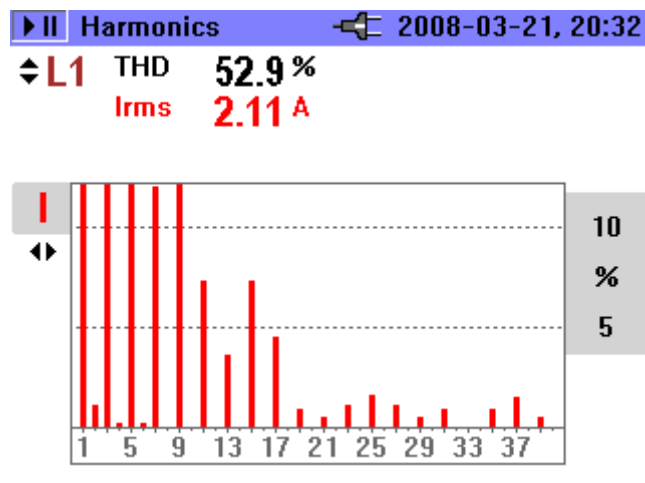**Fig. H-1 : Typical low-load domestic voltage and current waveforms**

Fig. H-2 : Typical low-load domestic current harmonic distortion