

# Fault Detection and Diagnosis of a Plastic Film Extrusion Process <sup>★</sup>

Sung-ho Hur<sup>\*</sup> Reza Katebi<sup>\*</sup> Andrew Taylor<sup>\*\*</sup>

<sup>\*</sup> *Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK (e-mail: sunghur@ieee.org)*

<sup>\*\*</sup> *DuPont Teijin Films UK Ltd, Dumfries, UK*

---

**Abstract:** This paper presents a new approach to the design of a model-based fault detection and diagnosis system for application to a plastic film extrusion process. The design constructs a residual generator via parity relations. A multi-objective optimisation problem must be solved in order for the residual to be sensitive to faults but insensitive to disturbances and modelling errors. In this paper, we exploit a genetic algorithm for solving this multi-objective optimisation problem and the resulting fault detection and diagnosis system is applied to a first-principles model of a plastic film extrusion process. Simulation results demonstrate that various types of faults can be detected and diagnosed successfully.

*Keywords:* Plastic film extrusion, Fault detection, Fault diagnosis, Parity relations, Genetic algorithms

---

## 1. INTRODUCTION

A fault is defined as an unpermitted deviation of at least one characteristic property or variable of the process (Isermann and Ball, 1996). Types of faults occurring in industrial processes include actuator faults, sensor faults, and component faults. For industrial processes to satisfy performance specifications, any faults should be detected and diagnosed correctly. This process can be realised by applying a fault detection and diagnosis (FDD) system, which can be data-driven, knowledge-based, or model-based. The FDD system developed in this paper is model-based. Unlike the data-driven and knowledge-based approaches (Chiang et al., 2001), the model-based approach exploits a first-principles model to generate residuals via consistency checks between the plant observations and a first-principles model. The three main ways to generate residuals are observers, parameter estimation, and parity relations.

In addition to the development of a FDD system, the results of this paper are applied to a first-principles model of a plastic extrusion process. This model is described in Hur et al. (2008) and is used to simulate the plant throughout this paper. This large-scale model consists of a number of smaller sub-models. Each of these sub-models describe a main stage of the plastic film extrusion process, such as extrusion, where polymer melt is fed into a die; casting, where the polymer melt is discharged through the die gap onto a rotating casting drum to form a continuous amorphous sheet; machine-directional (MD) stretching; cross-directional (CD) stretching; heat setting; and winding, where finished product is rolled.

Residual generation is the most important task in fault detection (FD) and is presented in Section 2. Optimal residuals require to be sensitive to faults but insensitive to disturbances, which leads to a multi-objective optimisation problem. Several analytical methods for solving this multi-objective optimisation problem have been suggested (Lou et al., 1986). By contrast, we break this tradition and utilise an evolutionary algorithm (Konak et al., 2006), or more specifically, a genetic algorithm, for solving this multi-objective optimisation problem in Section 3. By the use of a genetic algorithm, the possibility of finding the global optimisation solution – global minimum – is increased by avoiding the calculation of cost function gradients, which can lead to local minima. Another advantage is that it is relatively straightforward to implement.

Chen and Patton (1999) employed a combination of a genetic algorithm, the method of inequalities and the moving-boundaries algorithm for optimising an observer based residual generator. However, this combination has never been applied for optimising a parity relation based residual generator in any literature. The parity relation based residual generation has a few advantages over the observer based residual generation. Unlike the observer approach, the parity relation approach does not convert the reference model into the frequency domain, which can increase the computation time required for optimisation. Also, there is no need to employ an extra algorithm, such as the eigenstructure assignment method (Chen and Patton, 1999) for guaranteeing the stability condition. The plant observation for a plastic film extrusion process is the cross-directional (CD) thickness profile (Featherstone et al., 2000), which needs to be divided into many sections for controlling and monitoring purposes. When the observer approach is employed, the number of the residual signals equal the number of the CD sections and, therefore, determining the thresholds can be difficult or the use of a

---

<sup>★</sup> The authors gratefully acknowledge financial support from DuPont Teijin Films UK Ltd, UK EPSRC Industrial CASE Award, UK EPSRC Platform Grant (GR/R04683/01).

dimensionality reduction technique is required. However, when the parity relation approach is employed, all the residuals are arranged under one signal.

The application of this FD system to the first-principles model is presented with simulation results in Section 4. Once a fault has been detected, the next stage is to determine which fault has occurred and this stage is often referred to as fault diagnosis. We modify the fault detection (FD) system to a fault detection and diagnosis (FDD) system, and the application of this FDD system to the first-principles model is demonstrated in Section 5.

## 2. RESIDUAL GENERATION

The parity relation based residual generation requires a model in the state space form as follows:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2)$$

where  $\mathbf{y}(k) \in \mathbb{R}^n$  denotes the process measurement,  $\mathbf{u}(k) \in \mathbb{R}^n$  is the control action, and  $\mathbf{x}(k) \in \mathbb{R}^r$  represents the states.

The proposed FDD system design utilises the System Identification Toolbox<sup>TM</sup>7 in Matlab<sup>®</sup> to derive a state space model directly from the first-principles model, which we use to simulate the plant throughout this paper. The derivation is possible using standard Matlab functions. This state space model is linearised around the operating mean of the set-points because the process is usually assumed to be linear at the steady state during normal operation (Featherstone et al., 2000) and because the mean of the actual set-points equals the operating mean, which can range from 40 to 60%.

By substituting (2) into (1) from time instant  $k-s$  to time instant  $k$  to collect 's+1' samples of  $\mathbf{y}$  and  $\mathbf{u}$  as illustrated in Fig. 1, we have

$$\begin{bmatrix} \mathbf{y}(k-s) \\ \mathbf{y}(k-s+1) \\ \vdots \\ \mathbf{y}(k) \end{bmatrix} - \mathbf{H} \begin{bmatrix} \mathbf{u}(k-s) \\ \mathbf{u}(k-s+1) \\ \vdots \\ \mathbf{u}(k) \end{bmatrix} = \mathbf{W}\mathbf{x}(k-s) \quad (3)$$

where  $s$  denotes delay, and  $\mathbf{H}$  and  $\mathbf{W}$  are given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & 0 & \cdots & 0 \\ \mathbf{CB} & \mathbf{D} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{s-1}\mathbf{B} & \mathbf{CA}^{s-2}\mathbf{B} & \cdots & \mathbf{D} \end{bmatrix} \quad (4)$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^s \end{bmatrix} \quad (5)$$

To remove the non-measurable states,  $\mathbf{x}(k-s)$ ,  $\mathbf{p} \in \mathbb{R}^{n(s+1)}$  can be chosen such that

$$\mathbf{p}^T \mathbf{W} = 0$$

In turn, (3) becomes

$$\mathbf{p}^T \mathbf{Y}(k) = \mathbf{p}^T \mathbf{H}\mathbf{U}(k) \quad (6)$$

where

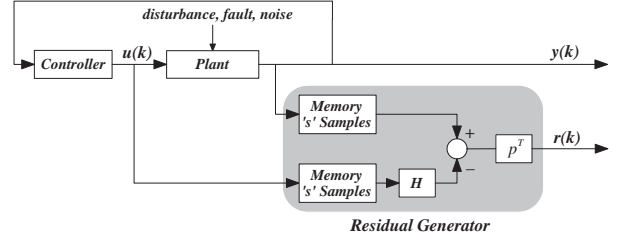


Fig. 1. Parity relation based residual generation

$$\mathbf{Y}(k) = \begin{bmatrix} \mathbf{y}(k-s) \\ \mathbf{y}(k-s+1) \\ \vdots \\ \mathbf{y}(k) \end{bmatrix} \quad (7)$$

$$\mathbf{U}(k) = \begin{bmatrix} \mathbf{u}(k-s) \\ \mathbf{u}(k-s+1) \\ \vdots \\ \mathbf{u}(k) \end{bmatrix} \quad (8)$$

leading to an equation for the residual,  $r(k) \in \mathbb{R}^1$  as follows:

$$r(k) = \mathbf{p}^T \mathbf{Y}(k) - \mathbf{p}^T \mathbf{H}\mathbf{U}(k) \quad (9)$$

However, (1) and (2) neglect unknown inputs, which include faults, noise, and disturbances. For the equations to approximate the process more closely, they can be modified to include the unknown inputs as follows:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{R}_1\mathbf{f}(k) + \mathbf{d}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{R}_2\mathbf{f}(k) \end{aligned} \quad (10)$$

where  $\mathbf{f}(k) \in \mathbb{R}^n$  denotes the fault vector and the fault distribution matrices  $\mathbf{R}_1$  and  $\mathbf{R}_2$  represent the influence of faults on the plant. These matrices can be determined if one has defined which faults need to be diagnosed.  $\mathbf{d}(k)$  can represent any other unknown inputs such as disturbances and modelling errors as follows:

$$\mathbf{d}(k) = \Delta\mathbf{A}\mathbf{x}(k) + \Delta\mathbf{B}\mathbf{u}(k) + \mathbf{E}\tilde{\mathbf{d}}(k) \quad (11)$$

where  $\mathbf{E}$  is a distribution matrix and  $\tilde{\mathbf{d}}(k)$  is the disturbance vector.  $\Delta\mathbf{A}$  and  $\Delta\mathbf{B}$  are the parameter errors or variations that represent process-model mismatch. Although, the noise term has been neglected here to simplify the algebra, it needs to be incorporated if the noise is significant when compared to the faults and disturbances and if the noise distribution matrix can be approximated.

Subsequently, (3) can be modified to

$$\mathbf{Y}(k) - \mathbf{H}\mathbf{U}(k) = \mathbf{W}\mathbf{x}(k-s) + \mathbf{L}\mathbf{N}(k) + \mathbf{M}\mathbf{F}(k) \quad (12)$$

where

$$\mathbf{N}(k) = \begin{bmatrix} \mathbf{d}(k-s) \\ \mathbf{d}(k-s+1) \\ \vdots \\ \mathbf{d}(k) \end{bmatrix} \quad (13)$$

$$\mathbf{F}(k) = \begin{bmatrix} \mathbf{f}(k-s) \\ \mathbf{f}(k-s+1) \\ \vdots \\ \mathbf{f}(k) \end{bmatrix} \quad (14)$$

where  $\mathbf{L}$  has a similar form to  $\mathbf{H}$  in which  $\{\mathbf{B}, \mathbf{D}\}$  are replaced by  $\{\mathbf{I}_n, \mathbf{0}_{n,n}\}$ , where  $\mathbf{0}_{n,n}$  is the  $n$ -by- $n$  zero

matrix and  $\mathbf{I}_n$  is the  $n$ -by- $n$  identity matrix. When  $\mathbf{B}$  is not a square matrix, zeros are appended below or to the right of  $\mathbf{I}_n$  appropriately. Similarly,  $\mathbf{M}$  is the same as  $\mathbf{H}$  except that  $\{\mathbf{B}, \mathbf{D}\}$  are replaced by the fault distribution matrices,  $\{\mathbf{R}_1, \mathbf{R}_2\}$ .

Substituting (12) into (9), the equation for the residual becomes

$$r(k) = \mathbf{p}^T \mathbf{Z} \mathbf{X}(k) + p^T \mathbf{M} \mathbf{F}(k)$$

where

$$\mathbf{Z} = [\mathbf{W} \mathbf{L}] \quad (15)$$

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{x}(k-s) \\ \mathbf{N}(k) \end{bmatrix} \quad (16)$$

Subsequently, two performance indices can be defined as follows:

$$J_1 = \|\mathbf{p}^T \mathbf{Z}\|_2 \quad (17)$$

$$J_2^{max} = \|\mathbf{p}^T \mathbf{M}\|_2 \quad (18)$$

where  $\|\cdot\|_2$  denotes  $L_2$  norm. Maximising  $J_2^{max}$  is equivalent to minimising

$$J_2 = -\|\mathbf{p}^T \mathbf{M}\|_2 \quad (19)$$

Finally, by minimising  $J_1$  and  $J_2$ , the residual can become sensitive to faults but insensitive to disturbances and modelling errors, which is the desired property of the residuals for FDD. The multi-objective optimisation technique used to minimise both performance indices at the same time utilises a genetic algorithm and is presented in the following section.

If enough information is given to determine  $\Delta \mathbf{A}$ ,  $\Delta \mathbf{B}$ , and  $\mathbf{E}$  in (11), these matrices can be incorporated into (10). If these matrices could be determined perfectly, the residual  $r(k)$  would be zero. However, these matrices cannot be determined for most real-life situations. Therefore, we also do not incorporate these matrices in this paper. More detailed discussion on determining the disturbance matrices can be found in Chapter 5 of Chen and Patton (1999).

### 3. MULTI-OBJECTIVE OPTIMISATION

A combination of the method of inequalities, the moving-boundaries algorithm, and a genetic algorithm is exploited for solving the multi-objective optimisation problem presented in Section 2.

#### 3.1 Method of Inequalities

The method of inequalities transforms the problem of the minimisation or maximisation of the performance indices to the problem of the satisfaction of a set of inequalities. That is, the problem becomes searching for a parameter set that satisfies the following inequalities:

$$J_i(\mathbf{p}) \leq \epsilon_i \quad (20)$$

where  $\epsilon_i$  ( $i = 1, 2$ ) is the bound on the performance index,  $J_i(\mathbf{p})$ , chosen by the designer. By restricting or relaxing the bound  $\epsilon_i$ , the designer can place a different emphasis. If  $J_1^*(\mathbf{p})$  and  $J_2^*(\mathbf{p})$  are the minimum values achieved, the designer should define  $\epsilon_i$  ( $i = 1, 2$ ) as

$$J_i^*(\mathbf{p}) \leq \epsilon_i \quad (21)$$

#### 3.2 Moving-boundaries Algorithm

To help solving the design problem presented above, Zakian and Al-Naib (1973) suggest an algorithm which they call the moving boundaries algorithm. The performance indices are first normalised as follows:

$$\phi_i(\mathbf{p}) = J_i(\mathbf{p})/\epsilon_i \quad (22)$$

In turn, the problem becomes satisfying

$$\phi_i(\mathbf{p}) \leq 1 \quad (23)$$

To solve (23), let  $\mathbf{P}_i$  be the parameter that satisfies the  $i^{\text{th}}$  performance index

$$\mathbf{P}_i = \{\mathbf{p} : \phi_i(\mathbf{p}) \leq 1\} \quad (24)$$

and  $\mathbf{P}$  be the parameter that satisfies both performance indices

$$\mathbf{P} = \left\{ \mathbf{p} : \max_{i=1}^2 \{\phi_i(\mathbf{p}) \leq 1\} \right\} \quad (25)$$

The search for an optimal  $\mathbf{P}$  can be achieved by solving the following optimisation problem:

$$\min \left\{ \max_{i=1}^2 \{\phi_i(\mathbf{p})\} \right\} \leq 1 \quad (26)$$

In order to solve (26), let  $\mathbf{P}^k$  be the parameter at step  $k$ , and define

$$\mathbf{P}_i^k = \{\mathbf{p} : \phi_i(\mathbf{p}) \leq \Delta^k\} \quad (27)$$

where

$$\Delta^k = \max_{i=1}^2 \{\phi_i(\mathbf{P}^k)\} \quad (28)$$

Now, let the problem become finding a new parameter  $p$  that reduces the largest performance index  $\Delta^k$  such that

$$\Delta^{k+1} \leq \Delta^k \quad (29)$$

The optimisation process terminates either when  $\Delta^k$  is less than 1 or when  $\Delta^k$  cannot be reduced further. If  $\Delta^k$  cannot be reduced further and persists being larger than 1, the appropriate bound should be relaxed. The most difficult part of this process is the provision of a trial parameter  $\mathbf{P}^{k+1}$  given  $\mathbf{P}^k$ . To solve this problem, many methods have been suggested since Zakian and Al-Naib (1973) introduced this technique, but especially the incorporation of a genetic algorithm has been proven to be effective and straightforward (Konak et al., 2006) and thus is a popular choice for multi-objective optimisation problems.

#### 3.3 Multi-objective optimisation via Genetic Algorithm

This paper assumes that the readers are familiar with genetic algorithms – detailed introduction to genetic algorithms can be found in books and papers such as Frenzel (1993)

The multi-objective optimisation procedures that utilise the combination of the method of inequalities, the moving boundaries algorithm, and a genetic algorithm for satisfying the performance indices are briefly summarised as follows:

*Step 1: Generation of Initial Population* The genetic algorithm generates an initial population. The size of the population is chosen by the designer.

*Step 2: Evaluation* The fitness functions (22) are evaluated using the population from *Step 1*. A score is then given using (28).

*Step 3: Reproduction* Using “ranking”, which is one of the available options (Frenzel, 1993), new children are created.

*Step 4: Elitism* A certain percentage of individuals in the current generation with the lowest fitness values are selected as “elites” and are passed on to form the population for the next generation. This stage ensures that the best individuals are not lost, but it can also increase the chance of being dominated by the elite individuals prematurely.

*Step 5: Recombination* Using “scattering”, which is one of the available options, cross-over children are created. This stage is motivated by the assumption that the best solution is contained in the population as a whole rather than in each individual and thus can be found by combining individuals.

*Step 6: Mutation* Mutation children are created by randomly changing the genes of parents’ individuals. We employ the Gaussian distribution to add a random vector to the parents. This stage is motivated by the probability that the initial random population did not contain all the information necessary to solve the problem and also that the individuals that were not allowed to reproduce offspring during the previous stages may have had some information necessary to solve the problem.

*Step 7: Termination Checking* Step 2 to 6 are repeated until either the following stopping criteria,  $\Delta^k \leq 1$ , has been met or it cannot be minimised further, in which case  $\epsilon_i$  ( $i = 1, 2$ ) should be relaxed.

### 3.4 Tuning of the genetic algorithm

In order to execute the genetic algorithm, the Genetic Algorithm and Direct Search Toolbox<sup>TM</sup>2 in Matlab® was utilised. For all the simulations illustrated in Sections 4 and 5, the tuning parameters were set as follows: population size: 20, number of generations: 100, reproduction method: ranking, elite count: 2 out of 20, cross-over fraction: 14 out of 20, cross-over function: scattering, mutation function: Gaussian, mutation fraction: 4 out of 20.

## 4. FAULT DETECTION

Having minimised both performance indices  $J_1$  and  $J_2$  in (17) and (19) at the same time using the multi-objective optimisation technique summarised in Section 3, the resulting FD system should produce a residual sensitive to faults but insensitive to disturbances. To assess the performance of this FD system, it has been applied to the first-principles model (Hur et al., 2008), which has been used to simulate the plant throughout this paper. Three of the simulations we have conducted are summarised as follows. We assume that only one fault or disturbance can occur at any time.

- *Sensor fault*: The sensor measurements are perturbed by  $\pm 15\%$  from 2000s
- *Disturbance 1*: Mass flow rate of polymer flowing into the die (Hur et al., 2008) are perturbed by  $\pm 10\%$  from 2000s
- *Disturbance 2*: Fast-roll speed is perturbed by  $\pm 10\%$  from 2000s

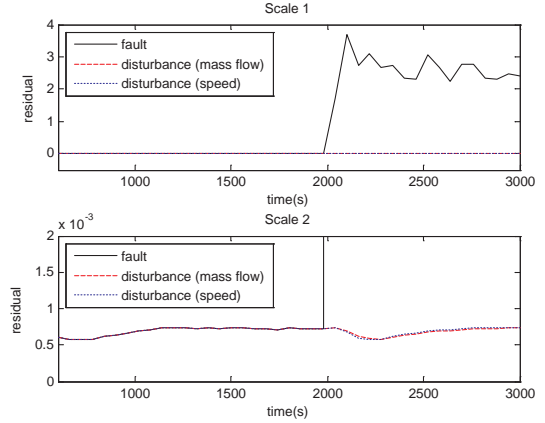


Fig. 2. Upper plot: residual when fault occurs at 2000s (solid); residual when mass flow rate is perturbed at 2000s (dashed); residual when fast-roll speed is perturbed at 2000s (dotted), Lower plot: same as the upper plot with a different scale

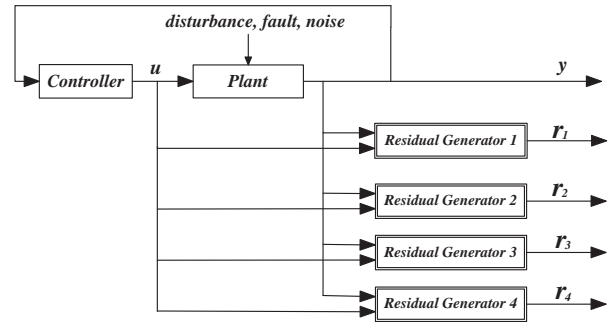


Fig. 3. FDD via Multiple Residual Generators

Distinguishing faults from disturbances is often controversial. Therefore, we define anything that can be rejected by the controller eventually as a disturbance and anything else as a fault. Furthermore, we assume that the plastic film is divided into 10 lanes in this paper (Hur et al., 2008).

Fig. 2 depicts the results for all these scenarios with one figure for comparison. The upper and lower plots are the same but have different y-axes. The results show that the residual is sensitive to the fault but insensitive to both disturbances, which is the desired property of a FD system. Although, this FD system is capable of detecting faults successfully, fault diagnosis has not been addressed. A FDD system conducts fault diagnosis after fault detection and is presented in the following section.

## 5. FAULT DETECTION AND DIAGNOSIS

The development of a FDD system is presented in this section. This FDD system is the same as the FD system demonstrated in Section 4 but has more than one residual generator, each sensitive to one specific fault, as shown in Fig. 3. Moreover, to make the scenarios more realistic, the mass flow rate of polymer flowing into the die is perturbed by  $\pm 10\%$  at all times – this is regarded as a disturbance which the FDD system requires to be insensitive to. Although this disturbance is persistent, we assume that only one fault occurs at any time

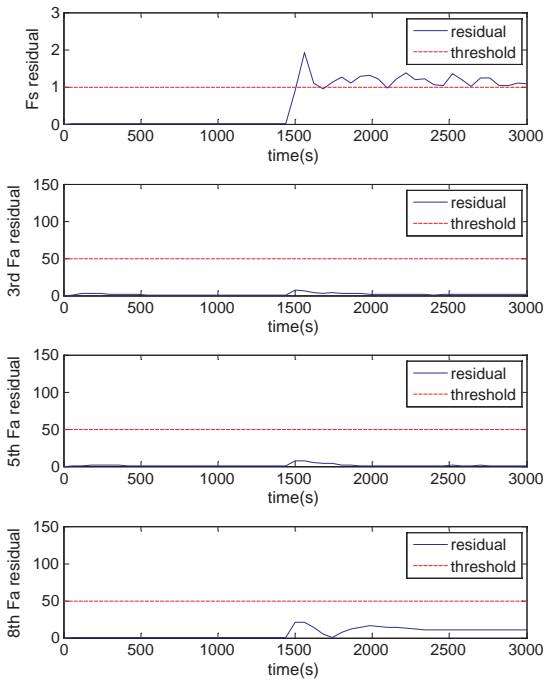


Fig. 4. Residuals when sensor fault occurs at 1500s

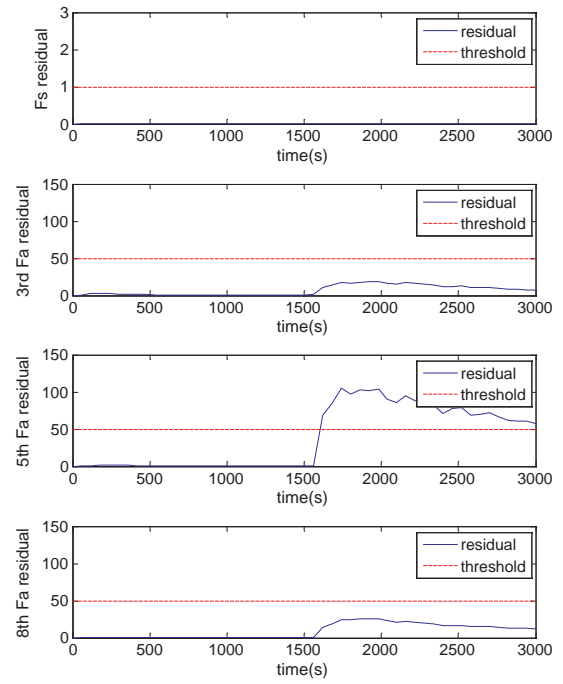


Fig. 6. Residuals when 5<sup>th</sup> die bolt becomes stuck at 1500s

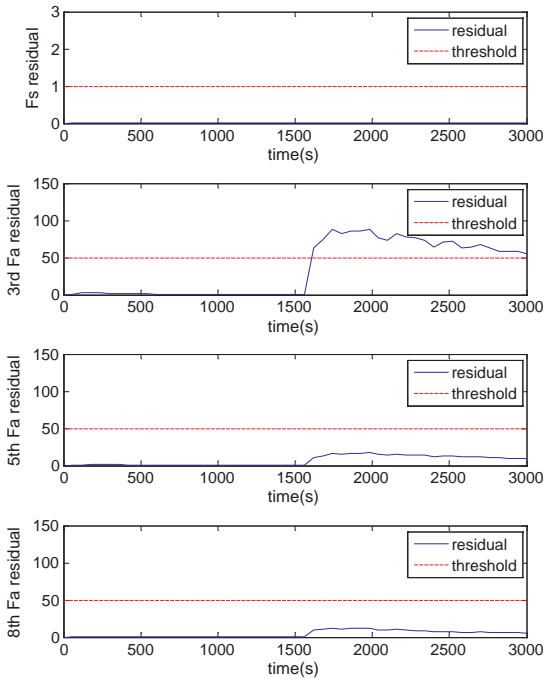


Fig. 5. Residuals when 3<sup>rd</sup> die bolt becomes stuck at 1500s

- *Sensor fault*: The sensor measurements are perturbed by  $\pm 15\%$  from 1500s
- *Actuator fault 1*: The third die bolt becomes stuck from 1500s

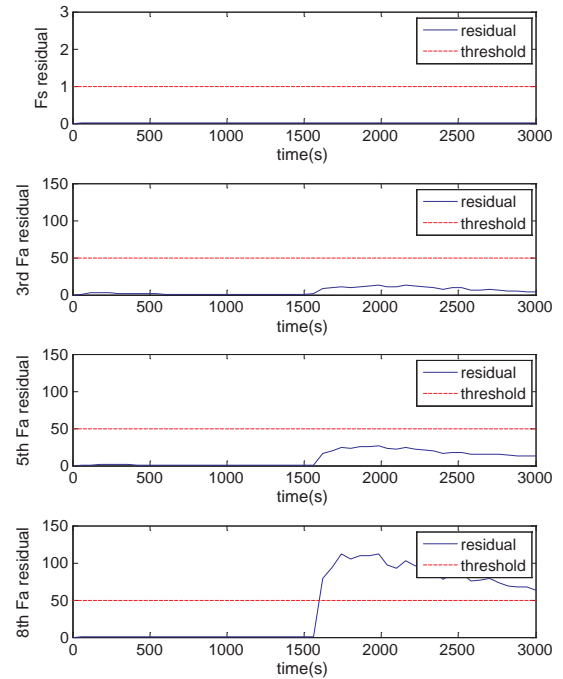


Fig. 7. Residuals when 8<sup>th</sup> die bolt becomes stuck at 1500s

- *Actuator fault 2*: The fifth die bolt becomes stuck from 1500s
- *Actuator fault 3*: The eighth die bolt becomes stuck from 1500s

## 6. CONCLUSION

If each residual generator can be made to be sensitive to one fault only, the faults can be diagnosed adequately. For each residual generator to produce a residual sensitive to a specific fault, each residual generator requires to have specific fault distribution matrices,  $\mathbf{R}_1$  and  $\mathbf{R}_2$  in (10). Consequently, each residual generator will have a different  $\mathbf{Z}$  and  $\mathbf{M}$  in (17) and (19), and  $\mathbf{p}$  will therefore be different for each residual generator.

For the first residual generator in Fig. 3 to be sensitive to the sensor fault, we approximate  $\mathbf{R}_1$  and  $\mathbf{R}_2$  as follows:

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{0} \\ \mathbf{R}_2 &= \mathbf{I}_n\end{aligned}\quad (30)$$

This intuitive approximation is widely accepted since sensor faults tend to affect the output equation without affecting the states (Chiang et al., 2001). The three residual generators are constructed to be sensitive to a fault occurring at a specific actuator. This is to demonstrate that the FDD system can be designed for a specific actuator among a few hundred actuators, thereby indicating which specific actuator has become faulty. However, since the sensing system employed by the first-principles model is a scanning gauge rather than an array of sensors (Featherstone et al., 2000), only one residual generator is constructed for the sensor fault.

For the residual generator to be sensitive to an actuator fault, we approximate  $\mathbf{R}_1$  and  $\mathbf{R}_2$  as follows:

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{B} \\ \mathbf{R}_2 &= \mathbf{D}\end{aligned}\quad (31)$$

where  $\mathbf{D}$  is assumed to be a zero matrix in this paper. This approximation is also widely accepted as the actuator faults affect the input vector  $\mathbf{u}(k)$  directly. Therefore, (10) becomes

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}(\mathbf{u}(k) + \mathbf{f}(k)) + \mathbf{d}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}(\mathbf{u}(k) + \mathbf{f}(k))\end{aligned}\quad (32)$$

For the residual generator to be sensitive to the  $i^{\text{th}}$  actuator fault, we let every entry of the matrix  $\mathbf{R}_1$  in (31) be a zero except  $i^{\text{th}}$  column – the  $i^{\text{th}}$  column of  $\mathbf{R}_1$  remains the same as the  $i^{\text{th}}$  column of  $\mathbf{B}$ . In this manner, the remaining three residual generators are constructed to be sensitive to the 3<sup>rd</sup>, 5<sup>th</sup>, and 8<sup>th</sup> actuator faults – 3, 5, and 8 have been chosen randomly to demonstrate that a residual generator can be designed for a specific actuator among as many as a few hundreds to indicate which actuator is faulty.

The results are depicted in Figs. 4, 5, 6, and 7. The thresholds were manually set at 2 for the first residual generator and 50 for the rest. If any of these thresholds is violated, a fault is detected. A fault can then be diagnosed by checking which threshold has been violated. Although only four residual generators were constructed in this paper, the number of residual generators will be considerably high in real-life depending on the number of faults required to detect and diagnose. From the experience, the plant personnel would know the types of faults which need to be detected and diagnosed. Using this information, the number and types of the residual generators can be determined appropriately. For good FDD, the design of accurate fault and disturbance matrices can play an important role.

This paper has reported on the design of a FDD system and its application to a first-principles model of a plastic film extrusion process. This model has been used to simulate the plant throughout this paper. The FDD system is based on the parity relations and thus requires solving a multi-objective optimisation problem. A genetic algorithm, which is an evolutionary algorithm, has been utilised for solving this multi-objective optimisation problem. This is a novel approach as an analytical method is usually utilised for solving this multi-objective optimisation problem. The simulation results in Section 4 demonstrated that the FDD system is sensitive to faults but insensitive to disturbances at the same time, which is the desired fault detection property. In addition to fault detection, the simulation results in Section 5 showed that fault diagnosis could also be conducted successfully. The design can be used for other kinds of processes, such as papermaking and steel-rolling as long as a reference model can be identified in the state space form.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge Prof. R. Patton at University of Hull and Dr. J. Chen at Brunel University for many useful discussions. Financial support of this project from DuPont Teijin Films UK Ltd, UK EPSRC Industrial CASE Award, and UK EPSRC is also gratefully acknowledged.

## REFERENCES

- Chen, J. and Patton, R.J. (1999). *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers.
- Chiang, L., Russell, E., and Braatz, R. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Advanced Textbooks in Control and Signal Processing. Springer.
- Featherstone, A., VanAntwerp, J., and Braatz, R. (2000). *Identification and control of sheet and film processes*. Advances in Industrial Control. Springer, London.
- Frenzel, J.F. (1993). Genetic algorithms. *IEEE Potentials*, 12(3), 21–24.
- Hur, S., Balderud, J., Katebi, R., and Taylor, A. (2008). A control and monitoring oriented model of a film manufacturing process. In *Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea*, 8357–8361.
- Isermann, R. and Ball, P. (1996). Trends in the application of model based fault detection and diagnosis of technical processes. In *Proceedings of the 13th IFAC World Congress*, volume N, 1–12. IEEE Press.
- Konak, A., Coit, D.W., and Smith, A.E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 91, 992–1007.
- Lou, X., Willsky, A.S., and Verghese, G.C. (1986). Optimally robust redundancy relations for failure detection in uncertain systems. *Automatica*, 22(3), 333–344.
- Zakian, V. and Al-Naib, U. (1973). Design of dynamical and control systems by the method of inequalities. *IEEE Control Theory & Application*, 120(11), 1421–1427.