

# Networked PID Control Design: A Pseudo-Probabilistic Robust Approach

L. F. Recalde, and R. Katebi

*Industrial Control Centre, University of Strathclyde, Glasgow, UK*  
*luis.recalde@eee.strath.ac.uk*  
*r.katebi@eee.strath.ac.uk*

---

**Abstract:** Networked Control Systems (NCS) are feedback/feed-forward control systems where control components (sensors, actuators and controllers) are distributed across a common communication network. In NCS, there exist network-induced delays in each channel. This paper proposes a method to compensate the effects of these delays for the design and tuning of PID controllers. The control design is formulated as a constrained optimization problem and the controller stability and robustness are incorporated as design constraints. The design is based on a polytopic description of the system using a Poisson pdf distribution of the delay. Simulation results are presented to demonstrate the performance of the proposed method.

---

## 1. INTRODUCTION

Networked Control Systems (NCS) are feedback/feed forward control systems where control components (sensors, actuators and controllers) are distributed across a common communication network. An NCS application offers low cost, ease of maintenance, flexibility, upgrading, redundancy and scheduling.

In NCS, there exist network-induced delays in each channel. Only some communication networks have been designed to meet real time constraints and hence are suitable for real-time application. Concepts such as data traffic, buffer contention, quantization constraints, protocols scheduling and synchronization, and transmission delays have to be considered in the design of NCS. More specifically, quality of service QoS in the communication network has to be achieved to avoid performance degradation in the control system. QoS is subject to transmission delays and data traffic. Transmission rates and packet size determine transmission delays. Medium access control sub-layer protocol MAC plays an important role in data traffic. The way information is transmitted through the network depends on deterministic or stochastic algorithms employed and once the information has been packed, there exist the possibility of dropping packets and missequencing.

Some studies analyse the dynamics of the network using time-delay systems theory. A naive approach may consider time-delays for synchronized processes; however synchronization in real distributed applications is extremely difficult [15]. The use of scheduling techniques to meet time constraints also introduces a varying sampling rate for each control loop [21].

Controller design depends on information availability. The controller design presented in this paper is subject to the following assumptions: the system is continuous with delayed inputs; the sensors are time-driven; the controller and actuators are event-driven; quantization constraints as well as varying sampling rates are not included. The controller is a discrete PID controller where its parameters are tuned using a

constrained optimisation method; and finally it can be proved that the closed-loop system is Hurwitz stable.

The structure of the paper is as follows: In section 2, a system description is presented for LTI systems with constant sampling rate. Some concepts for network-induced delay are introduced and delay distribution is defined as Poisson-wise. In section 3 a polytopic description of the model is formulated using the delay distribution. The set of models is reduced to a limited convex hull bounded by a tuning parameter. In section 4, a PID controller for the resulting set is tuned using constrained optimisation where performance and stability indices are incorporated as constraints. A numerical example is presented in section 5 to test the design approach.

## 2. SYSTEM DESCRIPTION

A typical NCS can be thought as two subsystems interacting across a common communication network.

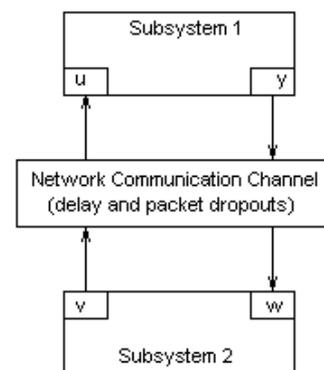


Figure 1. General framework for NCS

Subsystem 1 represents the plant, sensors, actuators and the quantizers. The system as seen by the controller side can have a discrete representation to match with a discrete controller. Subsystem 2 represents the controller.

The state-space realization of subsystem 1 and 2 are:

Subsystem 1:

$$\Sigma_1 = (A_p, B_p, C_p) \quad (1)$$

$$\begin{aligned} \dot{x}(t) &= A_p x(t) + B_p u(t) \\ y(t) &= C_p x(t) \\ t &\in \mathbf{R}^+ \end{aligned} \quad (2)$$

where  $u(t) \in \mathbf{R}^M$  are the system input signals,  $y(t) \in \mathbf{R}^R$  are the outputs and  $x(t) \in \mathbf{R}^N$  are the states of the system.  $A_p$ ,  $B_p$  and  $C_p$  are plant matrices of compatible dimensions.

For simplicity the plant is assumed to be linear and time invariant.

Subsystem 2:

$$\Sigma_2 = (E_c, F_c, G_c, H_c) \quad (3)$$

$$\begin{aligned} z(kh+h) &= E_c z(kh) + F_c w(kh) \\ v(kh) &= G_c z(kh) + H_c w(kh) \end{aligned} \quad (4)$$

where  $w(kh) \in \mathbf{R}^R$  are the discrete inputs,  $v(kh) \in \mathbf{R}^M$  discrete outputs and  $z(kh) \in \mathbf{R}^O$  the controller states.  $E_c$ ,  $F_c$ ,  $G_c$  and  $H_c$  are the discrete matrices of compatible dimensions.

$h$  represents the sampling time of the sensors. It is assumed to be constant for all sensors to facilitate the implementation of the controllers.

$$h_i = h \quad i = 1, 2, \dots, R$$

### 2.1. Network Induced Delay

The type of data transmitted across the bus architecture is countless number of small packets. In control applications the number of nodes that share the network can vary. The network can be dedicated for control purposes; or shared as in remote control applications over the Internet.

In any network, there is an effective transmission bandwidth. This bandwidth is defined as the maximum amount of meaningful data that can be transmitted per unit of time [14]. The utilization of this bandwidth depends on the packet size, the nodes requirements such as sampling rates and synchronized operations, and the MAC [14]. Not all network traffic is due to successful transmissions, collisions are usual among nodes attempting to transmit and the way the MAC deals with these collisions adds either a random or deterministic time-varying delay.

For control design, the delay has to be bounded. A practical assumption is to define this delay as a scalar nonnegative time-varying function of time for  $t \geq 0$  [28]. This is the network induced-delay  $\tau_k$ ,  $k = 1, 2, \dots$  in NCS. In our solution only network delay is considered and is split into input delay and output delay.

Input delay  $\tau_{CAk}$  is the time to transmit data between the controller and the  $i^{\text{th}}$  actuator. Assuming that the control signal is constant after the sample-and-hold device, i.e.  $f(t) = f(kh)$ ,  $kh < t < (k+1)h$ , the input delay is:

$$u(t) = v(kh - \tau_{CAk}), \quad (5)$$

The input delay is assumed to be bounded  $\tau_{CA(\min)} < \tau_{CAk} \leq \tau_{CA(\max)}$ , and the minimum delay can represent the inherent transmission delay.

Output delay  $\tau_{SCk}$  is the time to transmit data between the  $i^{\text{th}}$  sensor and the controller. Based on a constant sampling rate, the output delay in a sampling instant is:

$$w(kh) = y(t - \tau_{SCk}), \quad (6)$$

where  $\tau_{SC(\min)} < \tau_{SCk} \leq \tau_{SC(\max)}$  is bounded.

### 2.2. Ethernet network delay

The total time delay  $\tau$  to transmit data from the source node to the destination node can be expressed as three components [14]:

- i. the time at the source node,
- ii. the time at the network channel and,
- iii. the time at the destination node.

At the source node, the delay time consists of computation time and waiting time. Waiting time is critical in network traffic and consists of the time a message waits queuing in the buffer  $T_{queue}$  and the time a message waits once the node is ready to transmit  $T_{block}$ .

At the network channel, the time delay is a combination of the transmission delay and the propagation delay. This delay depends on the message size, data rate and length of the network cable [14].

Once the data has reached the destination node, there is a delay due to decoding and computation processes. The delays ( $\tau_{(\min)}$ ) at the network channel and at the destination channel can be calculated based on the network specifications. However, the delay at the source node and more specifically the delay, due to the waiting time is difficult to analyse. This delay is responsible for the time varying nature of the network.

Waiting time is the time a message has to wait before it is sent across the network. It depends on collisions, contention and transmission mechanisms and varies from network to network. The queuing time is the most difficult to determine, as it depends on the blocking time and periodicity of the messages. The blocking time is protocol dependent [24] and represents the way protocols manage transmissions and collisions.

In some cases, non-standard protocols can be used to discard old messages and set the queuing time to zero. Some authors have exploited this possibility by designing control-

oriented protocols such as TOD (try-once-discard) [15] and other dynamic bandwidth allocation methods.

The way each node access the network can be random or prioritised. For Ethernet-based networks the node that wants to transmit listens to the network and transmits once the network is idle. In case of collision, the transmitting node stops transmitting and waits  $x$  units of time to retransmit. This random time is determined by a Binary Exponential Back-Off algorithm.

After 16 attempts, the node stops transmitting and a report failure is sent. Hence the blocking time has a probabilistic behaviour [14] which can be described as follows:

$$\tau_k = E\{T_{block}\} = \sum_{j=1}^{16} E\{T_k^j\} + T_{resid} \quad (7)$$

where,  $T_{resid}$  is residual time seen by the node until the network is idle.

### 2.3. Delay Distribution

Under heavy traffic load, the time-varying nature of the delay can hardly be deterministic even for CAN-based networks [14]. In the worse scenario, a stochastic model of the network traffic can be formulated as a probabilistic process with known distribution. This distribution associates the probability that a particular delay happens.

As stated by Ryu in [19], Ethernet-based networks on Internet applications present Poisson-like network traffic under heavy traffic loads. In Pahjola's work [17] this distribution of the delay is considered as a Gamma distribution.

## 3. POLYTOPIC DESCRIPTION

For control design, both delays can be lump together as far as no packet dropouts are present. The resulting delay is still time-varying and random; however it can be introduced in the system model as input delay. The resulting model represents a rough approximation of network behaviour.

### 3.1. Time Delay Systems approximation

The input time delay modifies the state-space realization of the system as follows:

$$\begin{aligned} \dot{x}(t) &= A_p x(t) + B_p u(t - \tau_k) \\ y(t) &= C_p x(t) \quad t \in \mathbf{R}^+ \end{aligned} \quad (9)$$

where  $\tau_k = \tau_{sc,k} + \tau_{ca,k}$ .  $u(t) \in \mathbf{R}^M$  is the system inputs,  $y(t) \in \mathbf{R}^R$  is the outputs and  $x(t) \in \mathbf{R}^N$  is the states.  $A_p$ ,  $B_p$  and  $C_p$  are plant matrices of compatible dimensions.

It is possible to sample systems with delays when the control signal remains constant between sampling instants [2]. The resulting sampled-data system is finite dimensional, however due to the varying nature of the network delay, the sampled-data systems becomes time-varying.

The random nature of the delay makes the sampling process inaccurate. The randomness of the delay can be included by

least three different ways: a network induced-delay less than the sampling period, greater and multiple integer of the sampling period or simply greater than the sampling period. The last case involves the former two and can be assumed as the most general case.

The presence of network induced-delays greater than the sampling period is not very common. This case can be assumed to be similar to packet dropouts or unreliable communication with vacant sampling. Vacant sampling is defined as the absence of packets on the controller side during a sampling time. In this case, the controller uses the previous received packet or an interpolated value.

In our design the system with varying delays has a discrete realization based on time-driven sensors and event-driven actuators. For the maximum admissible delay, the discrete state-space realization of the sampled-data system is:

$$\begin{aligned} x(k+1) &= E_p(h)x(k) + F_{p0}(h, \tau_k)u(k - \alpha_k) \\ &\quad + F_{p1}(h, \tau_k)u(k - \alpha_k - 1) \end{aligned} \quad (10)$$

where  $\tau_k = \alpha_k h + \tau'_k$ ,  $0 < \tau'_k \leq h$ ,  $\alpha_k > 0$  and integer;

The above sampled-data system is time varying in  $F_{pi}$ ,  $i = 1, 2$ .

The resulting system represents a polytopic model subject to:

$$\Sigma_{d1}^i = [E_p(h), F_{pi}(h, \tau_k)] \in \Omega \quad (11)$$

and  $\Omega$  is the set,

$$\Omega = Co\{[E_{cl}(h), E_{pi}(h, \tau_1)], [E_{cl}(h), E_{pi}(h, \tau_2)], \dots, [E_{cl}(h), E_{pi}(h, \tau_k)]\} \quad (12)$$

with  $i = 1, 2$ ,  $k = 1, 2, \dots$  and Co referring to the convex hull and defined by linear models. The number of linear models in the set is not finite because the values of  $E_{pi}(h, \tau_k)$  constitute an approximation of the sampled-data NCS for random delays.

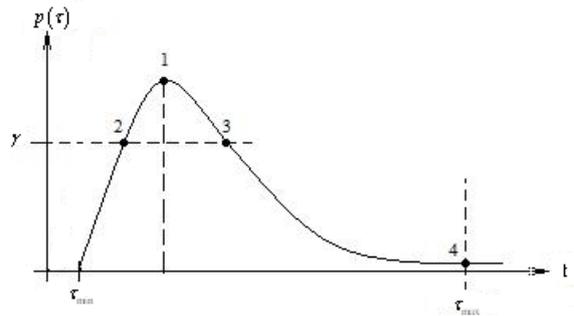


Figure 2. Delay Dependent set of Models

Using the information from the pdf of the delay,  $\Omega$  may be approximated to a finite number of linear models Fig. 2 [17]. For stability analysis the worse delay value can be used. It is, however, sufficient to include a set of models corresponding to the most probable values to achieve good performance and other values may be defined by a tuning parameter  $\gamma$ .

$\gamma$  is a sweeping factor in the pdf of the delay and represents a range of possible models with good performance for a given controller.

#### 4. CONTROLLER DESIGN

Most of the literature in PID controllers for NCS is based on either parameter optimisation [9], [17] and [24]; or adaptive parameter adjustment [7], [12] and [13], to meet desired design specifications. Sufficient conditions for stability have to be derived from maximum allowable delays.

In an optimal fashion, the controller design is jointly feasible if the closed loop system meets the design specifications [20]. From the literature of time delay systems, PID controllers have proved being stable when the parameters are subject to the delays [20].

For delay-dependent systems, the use of the derivative action is limited because linear extrapolation based on future values is not effective [1], but NCS is a particular case of systems with delays where control signals using prediction values can be suitable.

The controller to be implemented is a discrete PID controller of the form:

$$v(k) = v(k-1) + \sum_{i=0}^2 C_i e(k-i), e(k) = y_{sp}(k) - y(k) \quad (13)$$

where,  $y_{sp}(k)$  is the reference signal and admissible ranges of the controller parameters are:  $C_0 > 0$ ,  $C_1 < -C_0$  and  $-(C_0 + C_1) < C_2 < C_0$ . These ranges are aimed to ensure positive gain [11].

The controller parameters are sample rate-dependent. If the NCS model includes scheduling methods, the controller design leads to multi-rate digital controllers. In this paper, a single-rate controller design is sufficient to control the system.

##### 4.1 Pseudo-Probabilistic Robust Approach

Assuming that the set of polytopes is known, i.e.  $\Sigma_{d1}^i$ ,  $i=1,2,3$ , then two steps are necessary in the controller design. Firstly the convex optimisation problem must be solved for the model that represents the most probable delay  $\Sigma_{d1}^1$ . If performance specifications are satisfied for this model, it can be stated that there exists a controller that satisfies all the set, otherwise less conservative specifications have to be considered.

The other models depend on the tuning parameter  $\gamma$ . By sweeping the pdf, a range of possible models bounded by  $\{\Sigma_{d1}^2, \dots, \Sigma_{d1}^3\}$  can be generated. Thus performance specifications can be formulated using the closed loop model of combining  $\Sigma_{d1}^i$ ,  $i=1,2,3$  with the optimal PID controller  $\Sigma_2^{op}$ .

Stability conditions can also be specified using the fourth system model. In this study a fourth model is used to include the worse case model  $\Sigma_{d1}^4$ . If the closed loop system

represented by  $\Sigma_{d1}^4$  and  $\Sigma_2^o$  is stable, then stability is ensured for the full set.

The system is said to be stabilizable if and only if the following LMI is feasible, with P, Q and Q1 bigger than zero:

$$\Phi = \begin{pmatrix} -P+Q+Q_1 & 0 & 0 & (A-K)^T \\ 0 & -Q & 0 & (F_{p0}(\tau_i)K_a)^T \\ 0 & 0 & -Q_1 & (F_{p1}(\tau_i)K_a)^T \\ (A-K) & (F_{p0}(\tau_i)K_a) & (F_{p1}(\tau_i)K_a) & -P \end{pmatrix} < 0 \quad (14)$$

#### 5. NUMERICAL EXAMPLE

To illustrate the above design method the NCS is defined as a 10 node system connected across a communication network.  $\tau_k$  is the lumped delay and represents the round trip delay in Ethernet-based networks

Network parameters are taken from [14] to match with real values. The packet size is 8 bytes. Each node represents a simple first order system as follows:

$$x(k+1) = \begin{pmatrix} .99 & -.0006 & -.0001 \\ .0006 & 1 & 0 \\ & .0006 & 1 \end{pmatrix} x(k) + B_0(\tau_i)u(k-\alpha_i) + B_1(\tau_i)u(k-\alpha_i-1)$$

$$h = 572e-6$$

The total time to send 10 messages over the networks is [14]: 576  $\mu s$ . If the period is shorter than the total transmission time the traffic load increases and the network can become unstable. In the simulations, the sampling period will be equal to  $\tau_{(min)}$ , thus the network induced-delay is zero or  $\tau_k$ . The average time delay defined is 1267  $\mu s$  [14].

Table 1. System Coefficients

	$F_{p0}(h, \tau_k)$	$F_{p1}(h, \tau_k)$
Ethernet	$(-.9991, \dots, -.9997)$	$1e-3(.115, \dots, .213)$

Assuming a tuning factor  $\gamma$  corresponds to 70% of the probability of the delay, the resulting set is as follows:

Table 2. Controller Parameters

	$C_0$	$C_1$	$C_2$
Average Delay	.0074	.016	-.0105
Probability Delay	.0027	.0054	-.0037

The controller design is based on constrained optimisation using pattern search in MATLAB. Restrictions were defined as inequalities. The design is highly dependent on the initial conditions. Convergence of the controller parameter search was obtained only when the  $C_0$  was weighted ten times higher than the others. For other values no solution was obtained.

Simulations are based on a Simulink model using TrueTime1.5. This simulator allows co-simulation of controller-task execution in real-time kernels, network transmission and continuous plant dynamics [30]. Models 1 and 3 from Fig. 2 are tested. The three controllers are implemented as nodes in an Ethernet network with 10Mbps. The remaining 7 nodes are simulated as network traffic.

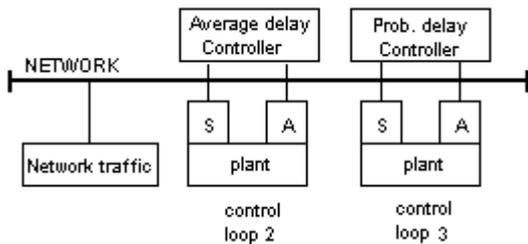


Figure 3. Control loops distribution

The delays used to obtain the models are simulated by changing the probability of packet lost. The following results were obtained:

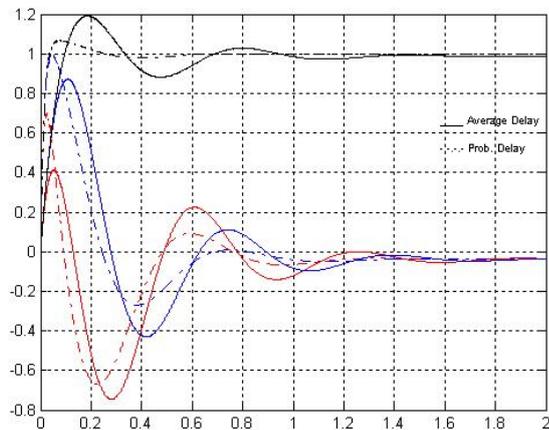


Figure 3. Output response for model 1

The simulations show better performance for average delay for model 1 where the correct delay is used. However, the performance of the proposed controller is better in model 3. For model 3, the controller based on Average delay tends to instability.

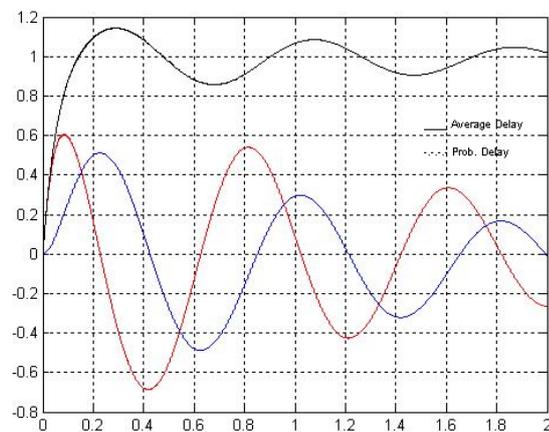


Figure 7. Output response for model 3

## 6. CONCLUSIONS AND FUTURE WORK

It was argued that to achieve good performance both network and system performance are to be considered. Furthermore, instability of the NCS is merely instability of the network due to heavy traffic. Network induced-delays have been lumped together for design purposes. The controller design is based on the expectation of the most probable delay accompanied by a set of models that depends on a tuning parameter  $\gamma$ . The method used to tune the controller can be extended to a probabilistic robust approach where the set of models will depend on random sampling of the pdf. Hence the controller design is not limited to a reduced range of models.

## 7. ACKNOWLEDGEMENTS

This work has been supported by the Programme Alþan, The European Union Programme of High Level Scholarships for Latin America, scholarship No.E07D402078EC.

The author expresses gratitude to Dr. Leonardo Giovanini for his technical comments and suggestions.

## References

- [1] Åström, K., Häggliund, T.: "The future of PID control", Control Engineering Practice 9, 2001.
- [2] Åstrom K. J., Wittenmark B.: "Computer controlled systems, Theory and Design", 1990
- [3] Bompert, V., Apkarian, P., Noll, D.: "Control design in time and frequency domain using nonsmooth techniques", Systems and Control Letters 57, 2008.
- [4] Dai, J., Cui, B.: "A new delay systems approach to quantized networked control systems", Manuscript draft, Jiangnan University, 2008.
- [5] Delchamps D. F.: "Stabilizing a Linear system with quantized state feedback", IEEE Transactions in Automatic Control, Vol. 35, No 8, August 1990.
- [6] Boyd, S., Barratt, C.: "Linear Controller Design: Limits of Performance", Prentice Hall, 1991.
- [7] Fang, L., Wu, Z.: "Fuzzy immune self regulating PID control for Networked Control Systems", IEEE, 2006.
- [8] Giovanini, L., Marchetti, J.: "Shaping Time-Domain responses with Discrete Controllers", Ind. Eng. Chem. Res., 1999.
- [9] Ghude, S.: "Design a PID Controller with Missing Packets in a Networked Servo-System", Master dissertation, March, 2007.
- [10] Harle, D.: Communication Network notes, University of Strathclyde, 2007.
- [11] Isermann, R.: "Digital Control Systems", Springer-Verlag, Berlin, 1981.
- [12] Kyung, C., Suk, L.: "Remote Controller Design of Networked Control Systems using Genetic Algorithm", IEEE, 2001.
- [13] Li, S., Wang, Z., Sun, Y.: "A novel Auto-tuning Robust PID controller for Networked Control Systems", IEEE, 2003.
- [14] Lian F., Moyne J., Tilbury D.: "Performance Evaluation Control Networks: Ehternet, ControlNet and DeviceNet", Technical report UM-MEAN-99-02, February 1999.
- [15] Lian F., Moyne J., Tilbury D.: "Analysis and Modelling of Network Control Systems: MIMO case with multiple delays", Proceedings of American Control Conference, Arlington, Virginia, June 2001.
- [16] Oppenheim A. V., R. Schafer W., Buck J.R.: "Discrete-Time Signal Processing", 1999.
- [17] Pahjola Michael: "PID Controller Design for Networked Control Systems", Master's thesis for the degree of Master of Science in Technology, Espoo, 9. January, 2006.
- [18] Polyanin, A., Mazhurov, A.: "Handbook of Mathematics for Engineers and Scientists", 2007.

- [19] Ryu, S., Cho, C.: "PI-PD-controller for robust and adaptive queue management for supporting TCP congestion control", Proceedings of the 37<sup>th</sup> Annual Simulation Symposium, IEEE, 2004.
- [20] Silva, G.: "PID controllers for Time-delay systems", 2004
- [21] Velasco, M., Marti, P., Villa, R., Fuertes, J.: "Stability of Networked Control Systems with Bounded Sampling Rates and Time Delays", IEEE, 2005.
- [22] Velasco, M., Marti, P., Frigola, M.: "Bandwidth management for distributed control of highly articulated robots", IEEE International Conference on Robotics and Automation, Barcelona, Spain, April, 2005.
- [23] Yang T. C.: "Networked Control Systems: a brief survey", IEE proceedings in Control Theory and Application, Vol. 153, July 2006.
- [24] Zhang, W., Branicky, M., Phillips, S.: "Stability of Networked Control Systems", IEEE Control Systems Magazine, February 2001.
- [25] Walsh G. C., H. Ye, Bushnell L. G.: "Stability analysis of networked control systems", IEEE Trans. Control Syst. Technol., pp. 438-446, 2002.
- [26] Kharitonov, V., Gu, K., Chen, J.: "Stability of Time-Delay Systems", 2003.
- [27] Low, S., Paganini, F., Doyle, J.: "Internet congestion control", IEEE Control systems Magazine, February 2002.
- [28] Halevi Y., Ray A: "Integrated communication and control systems, Part I-Analysis", Journal of Dynamic Systems, Measurements and Control, Vol. 110, pp. 367-373, December 1988.
- [29] Richard, J.: "Time-delay Systems: an overview of some recent advances and open problems", Automatica, Vol. 39, April 2003.
- [30] Ohlin, M., Henriksson, D., Cervin, A.: "True Time 1.5-Reference Manual", Department of Automatic Control, Lund University, January 2007.