Generalized Predictive Control in a Wireless Networked Control System

Joe Hsuan-Chiao Huang, Min-Fan Ricky Lee, Ming-Jong Tsai and Steven Fu-Hsin Chiu
Graduate Institute of Automation and Control
National Taiwan University of Science and Technology
rickylee@mail.ntust.edu.tw

Abstract - When a Networked Control System (NCS) operates over the communication network, one of the major challenges is the network-induced delay in data transfer among the controllers, actuators, and sensors. This delay may degrade system performance and even cause an unstable system. This paper proposes an adaptive Generalized Predictive Control (GPC) with the Kalman state estimator to cope with network-induced delay. The adaptive GPC is implemented in Wireless NCS (WiNCS) based on IEEE 802.11b, using the PiccSIM platform and Network Simulator version 2 (NS2) for simulation and analysis.

Index Terms - Networked control system; network-induced delay; packet loss; generalized predictive control; Network Simulator.

I. INTRODUCTION

Using data network, Ethernet cable, or wireless as the communication protocol in the feedback control loop has gained increasing attentions due to its cost effectiveness and flexibility. The Networked Control System (NCS) closes the feedback control loops through a network and the control signals to the actuators and the feedback signals from sensors are in the form of information packages [1-2].

Interconnecting the sensors, actuators, and controller via networks can eliminate wiring, reduce installation costs, and conduct remote monitoring and tuning. Further components and modules can be added without additional circuitry to the existing layout. The controllers can effectively share the data and the data can be further fused and integrated easily to the controller to make an intelligent decision or optimal operation in a large and complex process [3].

Dong et al. [4] modeled an NCS with both network-induced delay and packet loss in a transmission network. They proposed the feedback gain of a memory-less controller and derived the maximum allowable value of the network-induced delay by solving a set of linear matrix inequalities.

Garcia and Antsaklis [5] proposed the Model-Based Networked Control System which applies to a lifting process resulting in a Linear Time-Invariant and the asymptotic stability were analyzed.

A randomized multi-hop routing protocol causes time-varying delays for transmitting the sensor and control signal over the wireless network. Witrant et al. [6] proposed a predictive control scheme with a delay estimator, based on a Kalman filter with a change detection algorithm.

Loon and de Silva [7] proposed a GPC to compensate data-transmission delays incorporating a minimum-effort estimator to estimate missing or delayed sensor data and apply a variable-horizon adaptive GPC controller to predict the control actions to track a reference trajectory.

This paper proposes a wireless network control system algorithm and architecture based on the WLAN IEEE 802.11 technology. The Generalized Predictive Control (GPC) is applied to predict the network-induced delay and simulates it through the wireless network environment using the Network Simulator version 2 (NS2) in Linux. The PiccSIM is used as the platform in the client/server architecture for the WiNCS. The contribution of the proposed paper is summarized as

- The control system operation in communication networks and the NCS model with network-induced delay and packet loss using a general SISO NCS model is analyzed.
- A GPC control algorithm with the Kalman state estimator is implemented in WiNCS to reduce the network-induced delay effect.
- Experimental results confirm effectiveness of the proposed methodology.

II. ANALYSIS

The primary factors affecting NCS performance [8-10] are 1) Networked-induced delay, 2) Network scheduling, 3) Single-packet and multiple-packet transmission, 4) Packet loss and 5) Noise and Disturbance. The conventional control strategies are insufficient to deal with those issues and need alternative approaches for system modeling and analysis to address NCS characteristics and further construct an appropriate control strategy.

A. Networked-induced delay

Multiple time delay induced while the data transfer across the network among the controllers, actuators, and sensors as

- Propagation delay ($\tau_{\text{prop}}$): the data transmission time via wired or wireless networks is determined by the size of packet, bandwidth, and transmission distance.
- Switching delay ($\tau_{\text{swt}}$): the time required for data passing through electronic instruments such as a bridge, router, and switch. The instruments cannot send the data until all the packets have arrived.
- Access delay ($\tau_{\text{acc}}$): the time a network interface waits before it can access the network, mostly during packet collision or network congestion.
- Queuing delay ($\tau_{\text{que}}$): the time for newly arrived packets to wait until the previous packets have been sent. The packets will be queued by each switching device during the packet exchange in the communication network.
B. Delay Characteristic in NCS Model

Figure 1 shows the general NCS model with network-induced delay. There are two sources of network-induced delays from communication network, the sensors-to-controller, $\tau^{sc}$, and the controllers-to-actuators, $\tau^{ca}$. The controller processing delay $\tau^c$ is negligible in comparison to the network transmission delays.

$$\tau_k = \tau^{sc} + \tau^{ca}$$  \hspace{1cm} (1)

Figure 2 shows the network delay propagations diagram, where $u_k$ is the process input, $y_k$ is the process output, $k$ is the time index, and $T$ is the sampling time [12] as

$$u(t) = \begin{cases} u(k-1), & t_k < t \leq t_k + \tau_k \\ u(k), & t_k + \tau_k < t \leq t_k + T \end{cases}$$  \hspace{1cm} (2)

Figure 3 shows two types of packet loss in wireless loss.

Random uniform model is widely used for distributed loss and the Gilbert-Elliott model [13] for burst loss. The GE model contains two states of the Markov chain; the transmission channel is either available or not available, and named as “G (good)” or “B (bad)” respectively in Figure 4.

$$P_G \text{ and } P_B \text{ is the probability of packet loss occurring in the “G” and “B” state respectively. } P_{GB} \text{ and } P_{BG} \text{ is the probability of state change from “G” to “B” and “B” to “G” respectively in the transmission channel. If the transmission channel is in steady state, the probability of state change from “G” and “B” can be formulated as follows:}$$

$$\pi_G = \frac{P_{BG}}{P_{BG} + P_{GB}} \text{ and } \pi_B = \frac{P_{GB}}{P_{BG} + P_{GB}}$$

The average packet loss rate in the GE model is

$$P_{avg} = P_G \pi_G + P_B \pi_B \hspace{1cm} (3)$$

D. State Feedback Control of the NCS Model with Delay and Packet Loss

A linear, time-invariant, and continuous-time system and several assumptions are made to derive the NCS model as

$$\dot{x} = Ax(t) + Bu(t) \text{ and } y = Cx(t) \hspace{1cm} (4)$$

- Sensor is time driven with a fixed sample period of $T$
- Controller and actuator are event driven, and execute the calculation or operation immediately after receiving the data
- Single-packet transmission mode, the probability of packets out of order is zero.
- Packet loss rate is a constant
- The system plant is observable and the measurement noise is neglected
- The network-induced delay $\tau_k$ is less than the sample period $T$, i.e. $\tau_k \in [0, T]$

Figure 5 shows the system architecture of the NCS model with delay and packet loss. $u(k)$ is the system input (actuator input), $x(k)$ is the system output (sensor measurement), $\bar{x}(k)$ and $v(k)$ is the controller input and output [4,10, 14-15]. A switch $K_1$ and $K_2$ with a constant on/off rate is used to emulate the packet loss phenomenon in the communication network.

The NCS model with packet loss is constructed first without considering the network-induced delay. When the switches are on, the controller output equal to system input and the controller input equal to system output. When the switches are off, the system input and the controller input remain in the preceding state. The NCS model with packet loss is represented as follows as shown in Table 1.

$$S_1(K_1 \text{ On}): u(k) = v(k), S_1(K_1 \text{ Off}): u(k) = u(k-1) \hspace{1cm} (5)$$

$$S_2(K_2 \text{ On}): \bar{x}(k) = x(k), S_2(K_2 \text{ Off}): \bar{x}(k) = \bar{x}(k-1)$$

| TABLE 1 | SWITCH STATE V.S. NETWORK CONDITION |
The controlled system is rewritten from (1) and (4) as
\[
\dot{x} = Ax(t) + Bu(t - \tau_k) + e(t)
\]
(6)

The discrete-time equation is rewritten from (2) and (6)
\[
x(k+1) = \Phi x(k) + \Gamma_0 \tau(k) u(k) + \Gamma_1 \tau(k) u(k-1)
\]
(7)

where \( \Phi = e^{At} \),
\[
\Gamma_0 \tau(k) = \int_{t-k}^{t} e^{As} B \ ds \quad \text{and} \quad \Gamma_1 \tau(k) = \int_{t-k}^{t} e^{As} B \ ds
\]
(8)

The discrete-time model of the state feedback controller can be represented by \( v(k) = -K \dot{x}(k) \), where \( K \) is a gain matrix of the controller. From (7) and (8), let \( z(k) = [x(k), \dot{x}(k), u(k-1)]^T \), the state feedback of the NSC model with packet loss and delay is derived as follows
\[
z(k+1) = [\Phi_1 z(k) | i \in [1, A], i \in [1, I], i \in [1, I]]
\]
\[
\Phi_1 = \begin{bmatrix}
\Phi & -K \Gamma_0 \tau(k) & \Gamma_1 \tau(k) \\
0 & I & 0 \\
0 & 0 & I
\end{bmatrix}, \quad
\Phi_2 = \begin{bmatrix}
\Phi & -K \Gamma_0 \tau(k) & \Gamma_1 \tau(k) \\
0 & I & 0 \\
0 & 0 & I
\end{bmatrix}
\]
(9)

The switching probability of \( \Phi_1 \) for event \( i \) is according to the state of \( K_1 \) and \( K_2 \), as the packet loss rate in (3).

III. METHODOLOGY

When the control system operates over the networks, the network-induced delay affects system performance. The traditional controller generates a control signal at every sample time. If the network-induced delay exceeds one sample time, the new control signal cannot update the actuator which lead to an unstable control system. To avoid this situation, the controller generates one control signal series, including the predictive control signal, so the actuator can implement the backup control signal when network-induced delay occurs.

A. Adapter GPC

Assume a single-input single-output system operates around a specific set point after linearization. A predictive model known as the “Controlled Auto-Regressive Integrated Moving-Average” for Generalized Predictive Control as
\[
A(z^{-1})y(k) = B(z^{-1})u(k-1) + \frac{C(z^{-1})}{\Delta} e(k), \Delta = 1 - z^{-1}
\]
where \( y(k) \) is output signal, \( u(k) \) is input signal; \( e(k) \) is zero mean white noise. \( A, B, C \) are expressed as
\[
A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-na}
\]
\[
B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-nb}
\]
\[
C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_n z^{-nc}
\]
(10)

For simplicity, \( C(z^{-1}) \) polynomial is chosen to be 1. To enhance system robustness, the cost function includes the influence of \( u(k) \), and GPC algorithm apply control sequence to minimize a multistage cost function as
\[
J = \sum_{j=n_2}^{n_1} \delta(j)[\dot{y}(k+j) - w(k+j)]^2 + \sum_{j=1}^{n_1} \lambda(j) [\Delta u(k+j)]^2
\]
(11)

where \( \dot{y}(k+j) \) is the optimal j-step ahead prediction of system output, \( n_1 \) and \( n_2 \) are the minimum and maximum of the prediction horizons \( H_p : H_c \) is the control horizon; \( \delta(j) \) and \( \lambda(j) \) are weighting sequences. \( w(k+j) \) is the future reference trajectory as:
\[
w(k+j) = \alpha_j y(k+j) + (1-\alpha_j)y_r
\]
(12)

\( y(k) \) and \( y_r \) are set points and future outputs of the system respectively. \( \alpha \) is a parameter between 0 and 1 to adjust the system response (closer to 1, smoother response). Diophantine equation for predicting the preceding j-step output as
\[
1 = E_j(z^{-j}) A(z^{-1}) + z^{-j} F(z^{-1})
\]
(13)

where \( A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n+1} z^{-(n+1)} \). A state-space description [16-17] is given with dimension of the state vector as max (na+1, nb+1, nc)
\[
x(k+1) = Ax(k) + Bu(k) + \Pi u(k)
\]
(14)

\( y(k) = Cx(k) + v(k) \)
(15)

Where \( A = \\
\begin{bmatrix}
-a_1 & 1 & 0 & \cdots & 0 \\
-a_2 & 0 & 1 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
-a_n & 0 & 0 & \cdots & 0 \\
-a_{n+1} & 0 & 0 & \cdots & 0 \\
\end{bmatrix}, \quad B = \\
\begin{bmatrix}
b_0 \\
\cdots \\
\cdots \\
b_{nb-1} \\
\end{bmatrix}, \quad \Pi = \\
\begin{bmatrix}
c_1-a_1 & c_2-a_2 & \cdots & c_n-a_n & c_{nc}-a_{nc} \\
0 & 0 & \cdots & 0 & 0 \\
\end{bmatrix}, \quad C = [1 \ 0 \ \cdots \ 0 \ \cdots \ 0 \ 0 \ 0]
\]

where \( a_i \) is the coefficient of polynomial \( A \) given by (11). The random variable \( v \) and \( \omega \) represent disturbance input and measurement (sensor) noise and they are assumed as white Gaussian zero-mean with normal probability distributions.

\( x(k+1) = Ax(k) + Bu(k) \) and \( y(k) = Cx(k) \)
(16)

The z-domain transfer function \( R(z) \) and current output \( y(k) \) is obtained as
\[
R(z) = C(zI - A)^{-1}B = \frac{CB - \frac{CAB}{z^2} + \frac{CA^2B}{z^3} + \frac{CA^3B}{z^4} + \cdots}{z^n}
\]
(17)

\( y(k+1) = CB\Delta u(k-1) + CBA\Delta u(k-2) + \cdots \)
(18)

\( y(k+2) = CB\Delta u(k-1) + CBA\Delta u(k) + CBA_2 \Delta u(k-1) + \cdots \)
(19)

The output is obtained at the predictive horizon \( H_p = N \)
\[
y(k+n) = CB\Delta u(k+n-1) + CBA\Delta u(k+n-2) + \cdots \)
(20)

The prediction state of the system is then obtained as
\[
x(k+1) = Ax(k) + B\Delta u(k)
\]
\[
x(k+2) = Ax(k+1) + B\Delta u(k+1)
\]
\[ x(k+j) = A^j x(k) + \sum_{i=1}^{j-1} A^{j-i-1} B \Delta u(k+i) \]

A general term of \( y(k+j) \) with \( j = 1, 2, 3, \ldots, N \) as

\[ y(k+j) = \sum_{i=1}^{j} C A^{j-i-1} B \Delta u(k+j-i) + C A^i x(k) \]

The predictive output is \( \hat{y} = G \Delta U + f \), where

\[ \hat{y} = [\hat{y}(k+1) \hat{y}(k+2) \cdots \hat{y}(k+N)]^T \]

\[ \Delta U = [u(k) \ u(k+1) \cdots u(k+N-1)]^T \]

\[ f = [CA \ CA^2 \cdots CA^N]^T x(k) \]

\[ G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CBA & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CA^{N-N_u}B \end{bmatrix} \]

The cost function is applied to the control action \([18]\) and is formulated by penalization matrices, \( Q_y \) and \( Q_u \)

\[ J_{adp} = \sum_{j=1}^{N} \left\| [\hat{y}(k+j) - w(k+j)] Q_y \right\|^2 + \sum_{j=1}^{N} \| \Delta u(k+j-1) \|^2 \] \[ (16) \]

The cost function \((16)\) is rewritten as

\[ J_{adp} = (\hat{y} - w)^T \Delta U^T \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} (\hat{y} - w) \Delta U \]

Substitute predictive output \( \hat{y} \) into the cost function \( J_m \) as

\[ J_m = \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \Delta U - \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} (w - f) \] \[ (17) \]

To minimize the cost function \((17)\), the solution of the algebraic equation is the control action as follows

\[ \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \Delta U - \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} (w - f) = 0 \]

The above solution is re-organized as

\[ \bar{A} \Delta U = \bar{b} \] where \( \bar{A} = \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \) and \( \bar{b} = \begin{bmatrix} Q_y (w - f) \\ 0 \end{bmatrix} \]

(18)

The QR decomposition method based on the Householder algorithm is applied in the decomposition of matrix \( \bar{A} = QR \), where \( R \) is an upper triangular matrix and \( Q \) is an orthogonal matrix. \((18)\) is written as \( R^T Q^T \bar{b} - R \Delta U = 0 \) and the control signal is obtained as

\[ \Delta U = R^{-1} Q^T \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} (w - f) \] \[ (19) \]

\( \Delta U \) is the control signal for the whole predictive horizon \( N \), and the actual control signal is the first element in \((19)\).

**B. State estimator**

The optimal estimator to compute the state is based on the Kalman filter. The \( j \) step ahead system output is

\[ y(k+j) = CA^j \hat{x}(k) + \sum_{i=1}^{j} CA^{i-1} B \Delta u(k+j-i) \]

The estimation of the state vector \( \hat{x} \) can be obtained with the Kalman filter as follows

\[ \hat{x}(k) = A \hat{x}(k-1) + B \Delta u(k-1) \]

\[ + K_y(k) (y(k) - C [A \hat{x}(k-1) + B \Delta u(k-1)]) \]

where \( K_y(k) = P(k-1) C^T [CP(k-1)C^T + R]^{-1} \) is the Kalman filter gain matrix to adapt the estimation of model states to measure the controlled system outputs.

**C. Experimental setup**

The experimental hardware setup for WiNCS simulation is illustrated in Figure 6. Matlab/Simulink runs in Windows XP based client workstation for the emulation of the NCS Server/Client structure and network simulator version 2 (NS2) in Linux based server workstation for the emulation of the NCS wireless network environment. Client/ server is interconnected through an Ethernet cable and router for sharing the same simulation data.

![Fig. 6 Experimental Architecture](image-url)

The experimental setup is illustrated in Figure 7. To emulate the real NCS structure, the Server/Client module is setup through Matlab toolbox at the client workstation.

![Fig. 7 Software Architecture](image-url)

Table 2 presents the controller and actuator/sensor nodes configuration of IEEE 802.11b in NS2. When the network devices receive a large number of packets that exceed its maximum capacity, the newly arrived packets will be dropped until the network devices have a free queue space to accept incoming traffic. The AODV is implemented as a routing protocol algorithm in NS2, which builds a route to a destination only on demand.
The propagation model is a Two Ray Ground model [20] which considers transmission power consumed by a line-of-sight path between two mobile nodes and a ground reflection. The received power $P_r$ is presented as follows

$$P_r = \frac{P_t G_t G_r (h_t^2 - h_r^2)}{d^4 L}$$

Where $P_t$: Transmission power, $G_t$: Transmission antenna gain, $G_r$: Received antenna gain, $h_t$: Transmission antenna height, $h_r$: Received antenna height, $d$: Distance between sender and receiver and $L$: System loss factor. Table 3 shows the IEEE 802.11b simulation parameters in NS2.

| Parameters                      | Value                        |
|---------------------------------|------------------------------|
| Transmit/ Received Antenna Gain | $G_t$ and $G_r$, $1$          |
| System Loss Factor $L$          | $0.1 / 2.472$ GHz            |
| Data Rate (Bandwidth)           | $11$ Mb/ $0.0316$            |
| Collision Threshold/ Carrier Sense Power | $10.0/ 5.011872e-12$ |
| Received Power Threshold/ Rate for Data Frames | $5.82587e-09/ 11$ Mb |
| Rate for Control Frames         | $1$ Mb                       |

Figure 8 shows the wireless simulated environment in NS2. The Controller node (n0) and Actuator/Sensor node (n1) transmit the data packet via the IEEE 802.11b as an end-to-end transmission. The coordinates of the two nodes are (10, 10) of n0 and (60, 60) of n1, and each node propagation distance is 250 meters in the Two Ray Ground model.

The experiment implements the adaptive GPC with the state estimator, based on Kalman filter, and encapsulates the control signal and the sensor measurement into packets, transmitted in the wireless simulated environment IEEE 802.11b in NS2. Figure 9 shows the simulation architecture.

**Controller node**: The controller node contains the GPC controller, Kalman state estimator, sender, and receiver. The reference trajectory (controller input) is a square wave with a 15 sec. period and one peak between 0 to 100 secs.

**Actuator/Sensor node**: The actuator/sensor node includes the actuator, plant, sensor, sender, and receiver. The sensor measurement disturbance is implemented with a white noise distribution. A plant model in discrete state-space is

$$x(k+1) = \begin{bmatrix} -0.030 & -0.1 \\ 0.106 & 0.348 \end{bmatrix} x(k) + \begin{bmatrix} 0.212 \\ 1.383 \end{bmatrix} u(k)$$

$$y(k) = [0 \ 15.858] x(k)$$

**IV. RESULTS**

Figure 10 shows the system response with a sample time of 0.3 sec. Figure 11 and 12 shows the controller-to-actuator delay, sensor-to-controller delay, and sensor disturbance measurement respectively. The system is stable but with a higher overshoot and a longer settling time. Different sample times affect the system performance. For a system with a shorter sample time, the sender must generate more data packets. This might raise the packet loss rate and shorten the predictive horizon, which may make the system unstable.

**TABLE IV SIMULATION INFORMATION OF NODES**

| Simulation Information                      |
|--------------------------------------------|
| Number of packets generated                | Number of packets sent 668/668 |
| Average packet bytes                       | Number of bytes sent 43.77/44048 |
Figure 13, 14 and 15 shows the system response with a sample time of 0.2 sec. The system response is highly jittered with a longer settling time. Sender_c (n0) and Receiver_s (n1) dropped two packets and the dropped packet sizes were 132 bytes, presented in Table 5. In this situation, the system response could not follow the reference trajectory.

V. CONCLUSION

Network-induced delays in the wireless communication network are difficult to model, however this paper investigates the main problems that induce the time delay. With WiNCS simulated in NS2 using the Two Ray Ground model, this paper simplifies complex architectures in the wireless communication network for analysis proposes. First, this study implements WiNCS with the random delay to verify GPC controller capability with the Kalman state estimator to cope with time delay. Then WiNCS is implemented with NS2 to present the effect of different sampling times in the predictive horizon, namely system performance decreases when sample time decreases. When WiNCS is implemented with the sample time of 0.2 seconds, the packets start to drop, affecting system performance.

This paper proposes the WiNCS simulation on a low-level control system. Realizing WiNCS requires not only improving the control algorithm to compensate for time delay, but also improving wireless communication performance. The time delay generates when the packets exchange in the network. The algorithm for optimizing network performance communication is also important.

REFERENCES

[1] Z. Wei, et al., "Stability of networked control systems," Control Systems Magazine, IEEE, vol. 21, pp. 84-99, 2001.
[2] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems," Control Engineering Practice, vol. 11, pp. 1099-1111, 2003.
[3] F. Y. Wang and D. Liu, Networked Control Systems: Theory and Applications, 1 ed., Springer, 2008.
[4] E. Garcia and P. J. Antsaklis, "Model based networked control systems: a discrete time lifting approach," 2009.
[5] Y. Dong, et al., "State feedback controller design of networked control systems," Circuits and Systems II: Express Briefs, IEEE Transactions on, vol. 51, pp. 640-644, 2004.
[6] E. Witrant, et al., "Predictive control over wireless multi-hop networks," in Control Applications, 2007. CCA 2007. IEEE International Conference on, 2007, pp. 1037-1042.
[7] T. Poi Loon and C. W. de Silva, "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control," Control Systems Technology, IEEE Transactions on, vol. 14, pp. 707-718, 2006.
[8] W. Zhang, "Stability analysis of networked control system," Doctor of Philosophy Thesis, Department of Electrical Engineering and Computer Science, Case Western Reserve University, 2001.
[9] G. C. Walsh and Y. Hong, "Scheduling of networked control systems," Control Systems Magazine, IEEE, vol. 21, pp. 57-65, 2001.
[10] M. Garcia-Rivera and A. Barreiro, "Analysis of networked control systems with drops and variable delays," Automatica, vol. 43, pp. 2054-2059, 2007.
[11] K. C. Lee and S. Lee, "Performance evaluation of switched Ethernet for real-time industrial communications," Computer Standards & Interfaces, vol. 24, pp. 411-423, 2002.
[12] J. Nilsson, "Real-Time Control Systems with Delays," 1998.
[13] J.-P. Ebert and A. Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation," Telecommunication Networks Group, Technical University Berlin, 1999.
[14] Y. Liu, "Modeling and analysis of networked control systems with time delay and data packet dropout," Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 4609-4613, 2006.
[15] L. Sun and S. Guan, "A uniform modeling of networked control system with random delays," Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 4509-4512, 2006.
[16] D. W. Clarke, et al., "Generalized predictive control - part II. extensions and interpretations," Automatica, vol. 23, pp. 149-160, 1987.
[17] D. Soloway, et al., "GPC-Based Stable Reconfigurable Control," NASA Ames Research Center.
[18] K. Belda and J. Bohm, "Range-space predictive control for optimal robot motion," International Journal of Circuits, Systems and Signal Processing, Issue 1, Vol.1, 2007.
[19] M. Pohjola and S. Nethi, March, 2009, PiccSIM Manual http://autsys.dkk.fi/en/Control/PiccSIM.
[20] E. Altman and T. Jimenez, "NS Simulator for beginners," Dec.4, 2003.