# DESIGN AND IMPLEMENTATION OF NETWORKED PREDICTIVE CONTROL SYSTEMS

**S C Chai, G P Liu and D Rees**

*School of Electronics, University of Glamorgan, Pontypridd CF37 1DL, UK*

Abstract: This paper discusses the design and implementation of networked predictive control systems. A networked predictive control algorithm is proposed to compensate the network delay and achieve desired control performance of networked control systems. Two schemes of networked control systems are presented: one is the off-line simulation scheme and the other is a real-time application. Based on a networked control system test rig, the networked predictive control algorithm is implemented and applied to a practical servo control system. The simulation and practical experiment results illustrate the efficiency and feasibility of the proposed networked predictive control algorithm and simulation schemes. 

Keywords: Networked control, predictive control, simulation, implementation.

## 1. INTRODUCTION

The traditional communication architecture for control systems which has been successfully implemented in many areas for decades is a point-to-point architecture (the controller and implementer are at the same place) (F. Lian, *et al*., 2001). With the development of the internет and the diversity of physical setups, traditional control architecture is limited because of their modularity and centralisation of control. Networked control systems (NCS) has emerged as a significant topic for research as the result of the development of networks and appropriate control methodologies. Now, NCS can be found in manufacturing plants, aircraft, HVAC systems, automobiles and many other applications (L. G. Bushnell, *et al*., 2001). The main feature of NCS is the exchange, through communication networks, of system information and control signals between various physical components. So this type of system has the advantage of greater flexibility over traditional control systems, including greater flexibility in diagnosis and maintenance procedures . The studies on NCS cover very wide fields like medium of exchange, network protocols, network control methodologies, stability of NCS, scheduling of networked control systems (Walsh, G.C, *et al*., 2001).

In industrial control applications, there are three main buses used in NCS: the Ethernet bus, token-passing bus (e.g., ControlNet), and controller area network (CAN) bus (e.g., DeviceNet). Because of the world wide use of the Internet, the Ethernet network becomes the cheapest and widest medium of exchanging data. Ethernet uses a simple algorithm for operation of the network and has almost no delay at low network loads. Very little communication bandwidth is used to gain access to the network compared with the token bus or token ring protocol. Ethernet used as a control network commonly works at the 10 Mb/s standard (e.g., Modbus/TCP) and at high speed transmission rates of 100Mb/s to 1GMb/s.

There are two protocols, UDP and TCP, which are used for Ethernet networks. The UDP protocol is a connectionless protocol that runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offers a direct way to send and receive datagram over an IP network. TCP however adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received, which is more time consuming.Since UDP has two critical advantages over TCP, namely speed and overhead, it is the protocol that is normally used in NCS.

Although the NCS has many potential advantages over existing technology, there exists several control problems. These problems are network delay, data dropout, sampling and transmission method, and they are not easy to overcome using conventional control methods. To solve these problems, many methods have been adopted, such as the augmented deterministic discrete-time model method (Tham, M.T, *et al*., 2002) and optimal dropout compensator method (W. K. Ho,, *et al*., 2001). But, these methods have put some strict assumptions on NCS, e.g., the network time delay is less than a sampling period, or they have resulted in a solution that is not practical to implement. Liu et al(G P Liu,, *et al*., 2004) have proposed a networked predictive control method where they have considered the stability of the closed-loop NCS. This method can be readily implemented and actively compensates for the network time delay.

Now there are a number of research papers on network control systems (Tham, M.T, *et al*., 2002; W. K. Ho,, *et al*., 2001,and (G P Liu,, *et al*., 2004) but the majority focus on control theory and system stability analysis. The implementation aspects of these methods in the main are not addressed on those papers. In this paper we will seek to address these shortcomings by designing a controller and implementing the control methodology within a network structure. This paper is organised as follows: section 2 describes the basic idea of the networked predictive control method; Section 3 gives the simulation of networked control systems. Section 4 is the application example of the networked predictive control algorithm to a networked servo control system. Finally, some conclusions are made.

## 2. NETWORKED PREDICTIVE CONTROL

### *2.1 Networked predictive control scheme*

The networked predictive control (NPC) scheme was proposed by Liu *et al*(G P Liu,, *et al*., 2004) . However in this paper the time delay in the feedback channel is not taken into account or was taken as a constant value. But in reality the network delay in the feedback channel is time-varying rather than constant. But it is reasonable to assume that the time delays in the forward channel and feedback channel are the same and are not changing very fast. The structure of NPC is shown in Figure 1.

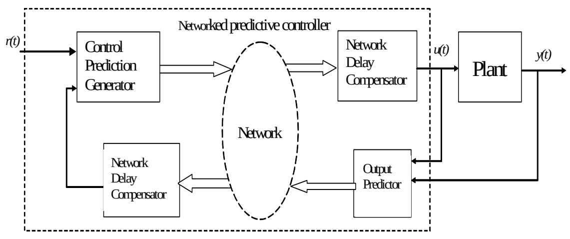


Fig. 1 The structure of NPCS

The NPC scheme mainly consists of a control prediction generator, an output predictor and forward and feedback delay compensators.

The control prediction generator is designed to generate a set of future control sequences. The forward and feedback delay compensators are used to compensate for the unknown random network delay in the forward and feedback channel. The output predictor is to create a set of plant output prediction sequences. In the NPC scheme, all predictive control sequences at one time are packed and sent to the plant side through a network. The network delay is compensated by choosing the latest control value from the control prediction sequences that are available on the plant side. For example, if the following predictive control sequences are received on the plant side:

$$\begin{bmatrix} u(t-k\,|\,t-k) \\ u(t-k+1\,|\,t-k) \\ \vdots \\ u(t\,|\,t-k) \\ \vdots \\ u(t-k+N\,|\,t-k) \end{bmatrix} \qquad (1)$$

where $u(t|t\text{-}k)$ is the control predication for time $t$ at time $t\text{-}k$. The output of the forward delay compensator will be

$$u(t) = u(t\,|\,t-k) \qquad (2)$$

which is the latest predictive control value for time t . Similarly, in the feedback channel, the output prediction sequence

$$\begin{bmatrix} y(t-k\,|\,t-k) \\ y(t-k+1\,|\,t-k) \\ \vdots \\ y(t\,|\,t-k) \\ \vdots \\ y(t-k+N\,|\,t-k) \end{bmatrix} \qquad (3)$$

is packed and transmitted to the controller side. So, the output of the feedback delay compensator is

$$\bar{y}(t) = y(t\,|\,t-k) \qquad (4)$$

which is the prediction of the system output for time t.

### *2.2 Implementation of networked predictive control*

Consider a single-input single-output discrete-time plant described by

$$y(t) = G(z^{-1})u(t-1) \qquad (5)$$

where the transfer function $G$ is

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} \qquad (6)$$

*y(t)* and *u(t)* are the output and control input respectively, *A* and *B* are polynomials. The networked predictive control scheme can be implemented in the following steps:

The first step is to identify the model of the plant to be controlled. Since the performance of the predictor

highly depends on the model accuracy. The model must fit the plant as closely as possible. In this paper, it is assumed that the model is the same as the plant.
The second step is to design a controller for the system without network time delay to satisfy the desired dynamic and static control requirements of the system. Any conventional and advanced control method can be used, such as PID, LQG and robust control methods etc. So, it is assumed that the controller is represented by the following transfer function:

$$L(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} = \frac{d_0 + d_1 z^{-1} + \cdots + d_p z^{-p}}{1 + c_1 z^{-1} + \cdots + c_q z^{-q}} \quad (7)$$

Thus, the output of the controller is

$$\bar{u}(t) = L(z^{-1})(r(t) - \bar{y}(t)) \quad (8)$$

where $\bar{u}(t)$ is the output of the controller, $r(t)$ is the reference input, $\bar{y}(t)$ is the output prediction of the plant The controller can be expressed by

$$\bar{u}(t) = u(t|t) = -\sum_{i=1}^{p} c_i \bar{u}(t-i) + D(z^{-1}) r(t) - \sum_{i=0}^{q} d_i \bar{y}(t-i) \quad (9)$$

The third step is to design the control predictor to generate predictive control sequences. The Diophantine equation method has been widely adopted. However that method is not easy to extend and program, particularly in the case of a random network delay. Here, a recursive method will be used. If the model is the same as the plant and there is no network time delay, the predictive output of the plant at time *t* can be obtained by

$$\bar{y}(t) = G(z^{-1}) \bar{u}(t-1) \quad (10)$$

So, the output prediction for time *t+1* can be expressed by

$$\bar{y}(t+1|t) = -\sum_{i=1}^{n} a_i \bar{y}(t-i+1) + \sum_{i=0}^{m} b_i \bar{u}(t-i) \quad (11)$$

Based on equation (9), the predictive output for time t*+1* can also be given by

$$\begin{aligned} u(t+1|t) = &-c_1 u(t|t) - \sum_{i=2}^{p} c_i \bar{u}(t-i+1) + D(z^{-1}) r(t+1) \\ &- d_0 \bar{y}(t+1|t) - \sum_{i=1}^{q} d_i \bar{y}(t-i+1) \end{aligned} \quad (12)$$

Then, at time $t+j$ the output prediction and control prediction on the controller side can be obtained as follows:

$$\begin{aligned} \bar{y}(t+j|t) = &-\sum_{i=1}^{\min\{n,j-1\}} a_i \bar{y}(t+j-i|t) - \sum_{i=j}^{n} a_i \bar{y}(t+j-i) \\ &+ \sum_{i=o}^{\min\{m,j-1\}} b_i u(t+j-i-1|t) + \sum_{i=j}^{m} b_i \bar{u}(t+j-i-1) \end{aligned} \quad (13)$$

$$\begin{aligned} u(t+j|t) = &-\sum_{i=1}^{\min\{p,j\}} c_i u(t+j-i|t) - \sum_{i=j+1}^{p} c_i \bar{u}(t+j-i) + D(z^{-1}) r(t+j) \\ &- \sum_{i=0}^{\min\{q,j-1\}} d_i \bar{y}(t+j-i|t) - \sum_{i=j}^{q} d_i \bar{y}(t+j-i) \end{aligned} \quad (14)$$

for $0 < j \le N$.

Summarising the above procedure leads to the following implementation algorithm for the forward channel.

(1) Given $\bar{y}(t-i)$ and $\bar{u}(t-i-1)$, for *i=0, 1, 2, …*. Let *j=0*;
(2) Calculate $\bar{u}(t+j|t)$ using (12);
(3) Calculate $\bar{y}(t+j+1|t)$ using (13);
(4) Set *j=j+1*;
(5) If *j<=N*, go back to step (2), otherwise, stop.

Clearly, the above algorithm gives

$$\left[u(t|t)\, u(t+1|t) \cdots u(t+j|t) \cdots u(t+N|t)\right]^T \quad (15)$$

For the feedback channel, the implementation algorithm is similar to the above. Based on given $y(t-i)$ and $u(t-i-1)$, for *i=0, 1, 2, …*, the following output predictions can be obtained.

$$\left[y(t|t) \quad y(t+1|t) \cdots y(t+j|t) \cdots y(t+N|t)\right]^T \quad (16)$$

The forward and feedback delay compensators are implemented as

$$u(t) = u(t|t-k) \quad \text{and} \quad \bar{y}(t) = y(t|t-k) \quad (17)$$

Therefore, the networked predictive control scheme can be implemented.

## 3. SIMULATIONS OF NETWORKED CONTROL SYSTEMS

### *3.1 Synchronisation of Networked Control Systems*

In networked control systems, one important issue is the synchronization of the whole system. For the sake of simplicity, the following assumptions have been made for the synchronization of NCS:
1) The network delays in the forward channel and feedback channel are the same.
2) The network delays do not change very quickly.
3) The forward delay compensator is located on the controller board.
To assess the forward and feedback delay, in order to design a delay compensator a measurement sinusoidal signal is transmitted over the network control system. Using the current measure signal value $d(t)$, the forward path delay compensator can calculate one-step, two-step …. and *N*-step time delay estimate of a measured signal $d(t-1), d(t-2) \ldots d(t-N)$. Comparing these values

with the returned value of the measured signal which was originally sent from the forward delay compensator, the total the network delay (including forward and feedback channels) is calculated. The forward and feedback time delays are then assessed to be half the total network delay.

### 3.2 Off-line Simulation

An off-line simulation scheme is presented for the networked predictive control system, as shown in Figure 2.

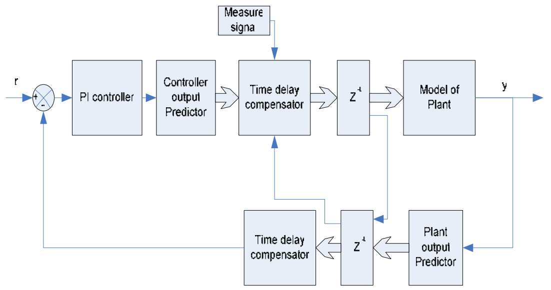


Fig. 2 Control scheme for simulation

To show the operation of this scheme, an off-line simulation scheme was implemented using the following model of a servo control system:

$$G(z^{-1}) = \frac{-0.00866z^2 + 1.27822z}{z^2 - 1.66168z + 0.658009} \tag{18}$$

Two cases of the network delay are simulated: one is the constant delay and the other is the random delay. The simulation results of the networked predictive control system for the cases of 1-step, 2-step and 3-step constant network delay in both forward and feedback channels is shown in Figure 3. It is clear from the results that the control performance of the closed-loop system for those three different network delays is the same. This means that the networked predictive control scheme can actively compensate for the network delay.

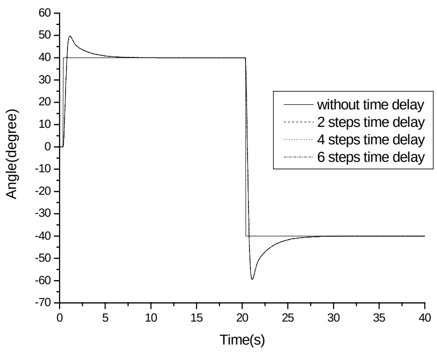


Fig.3 The responses of the closed-loop NPCS with different simulated constant network delays

To simulate a random network delay, a random sequence is employed, which is shown in Figure 4. The responses of the closed-loop NPCS with the above random delay and without time delay are given in Figure 5. Clearly, the NPCS also has good control performance for random time delay.

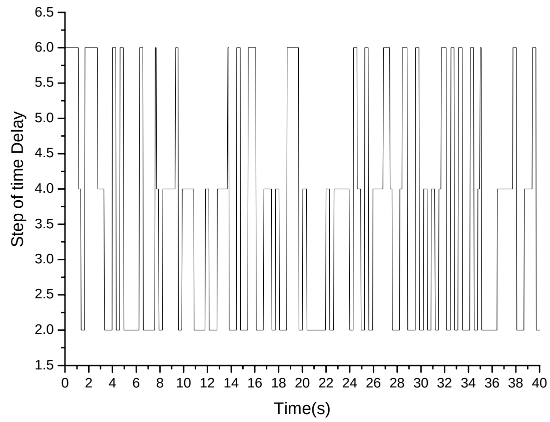


Fig. 4 The random time delay sequence

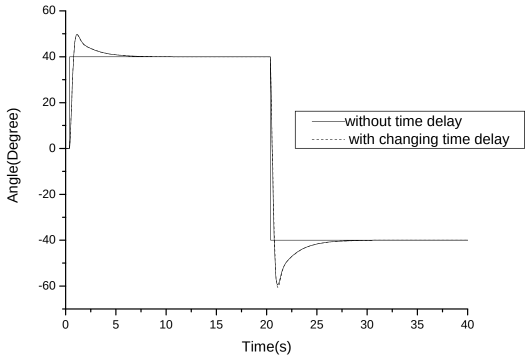


Fig. 5 The responses of the closed-loop NPCS with simulated random network delay

### 3.3 Real-time simulation

The real-time simulation is that the control program runs in a real-time embedded microprocessor system, where the plant to be controlled is still a mathematical model. A real-time simulation scheme for the networked predictive control system is proposed, which is shown in Figure 6. It consists of the controller part and the simulated plant part.

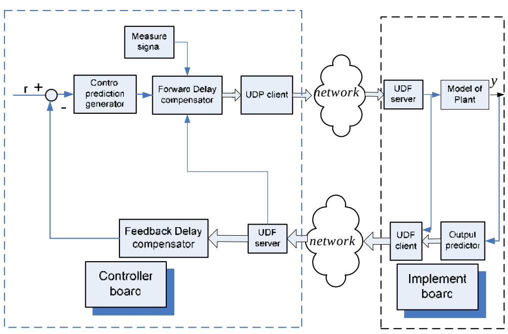


Fig. 6 Block diagram of real time simulation

For the implementation of the real-time simulation scheme, uClinux is chosen as the operating system of the real-time embedded microprocessor system.

uClinux is equipped with a full TCP/IP stack and is an internet-ready OS for embedded systems. The networked predictive control strategy is realised in Simulink. The controller part and simulated plant part are designed in two individual Simulink blocks. Then, the Real-Time Workshop in Matlab which generates, cross-compile and link program codes from Simulink (MathWork, *Real-Time Workshop for Use with simulink*. 2004), is adopted to create executable codes for the controller and simulated plant parts. Finally, these executable codes are downloaded to two real-time embedded microprocessor systems which are connected by Ethernet.
For the real-time simulation, the plant and controller are the same as those used for the off-line simulation. The real-time simulation results of the networked predictive control system are shown in Figure 7.

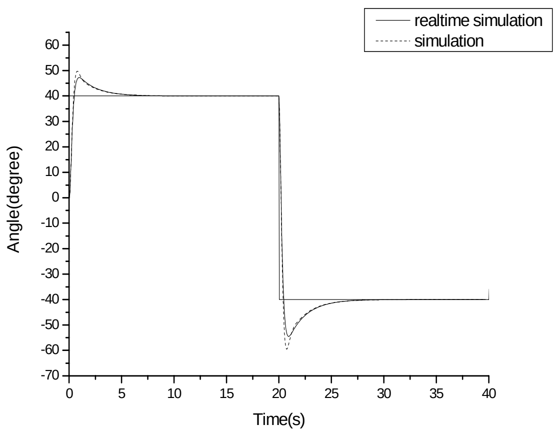


Fig .7 The responses of real-time simulation

Because the network delays in the forward and feedback channels are not the same in the real network, the network delay cannot be compensated for exactly. However, NPC can still achieve a similar control performance to one of NPCS without network delay.

## 4. APPLICATION TO A NETWORKED SERVO CONTROL SYSTEM

### *4.1 Networked control system test rig*

To apply the networked predictive control strategy to practical systems, a networked control system test rig is built. This rig consists of a networked control board (as shown in Figure 8), networked implement board and a servo control plant (as shown in Figure 9). The kernel chip of the networked control and implement boards is Samsung's S3C4510B. It is a cost-effective, high-performance microcontroller solution for Ethernet-based systems. The integrated Ethernet controller S3C4510B is designed for use in managed communication hubs and routers . The S3C4510B can operate at either 100-Mbits or 10-Mbits per second in half duplex or full-duplex mode.

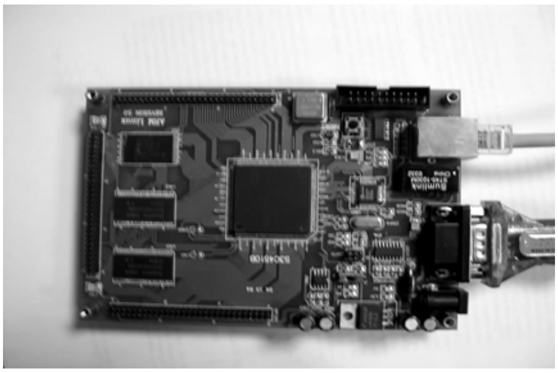

Fig. 8 The networked controller board

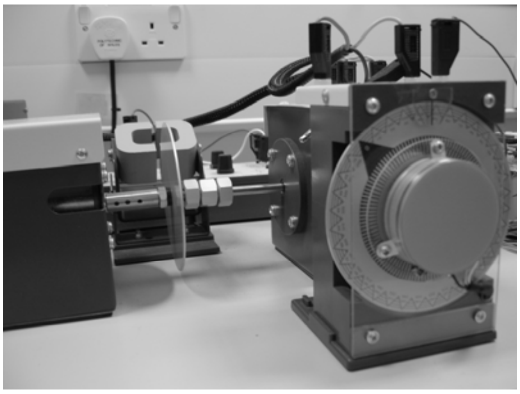

Fig. 9 The networked implement board and servo control system

The implementation board has eight 12-bit A/D input channels and two 16-bit D/A channels. The plant to be controlled is a position servo control system.

### *4.2 Practical experiments*

The block diagram of the network based servo control system is shown in the Figure 10. The transfer function of this servo control system was identified using the least squares method and is given in (18).

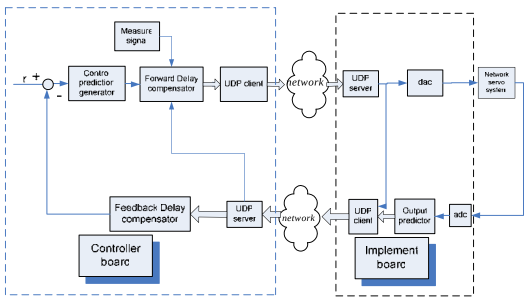


Fig. 10 The networked servo control system

The real networked predictive servo control system uses a PI controller, where the proportional gain is 2.6 and the integration gain is 1.2. For the sampling rate of 0.04s, the responses of the real closed-loop networked servo control system are shown in Figures 11-13.

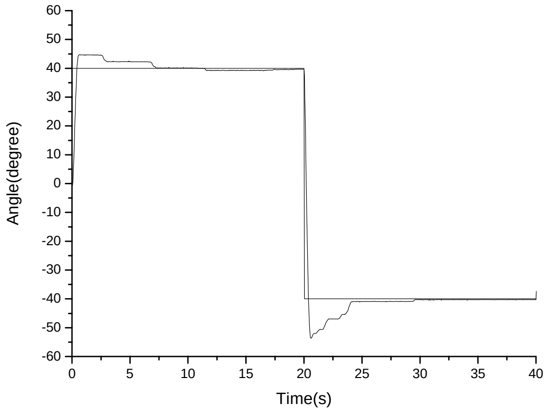


Fig.11 The response of the system without network Delay

The response of the system without network delay (*i.e.*, no network is used in the system) is shown in Figure 11. The response of the networked control system with network delay is shown in Figure 12, where no network delay compensator is used. It shows that the networked control system has poor control performance with network time delay.

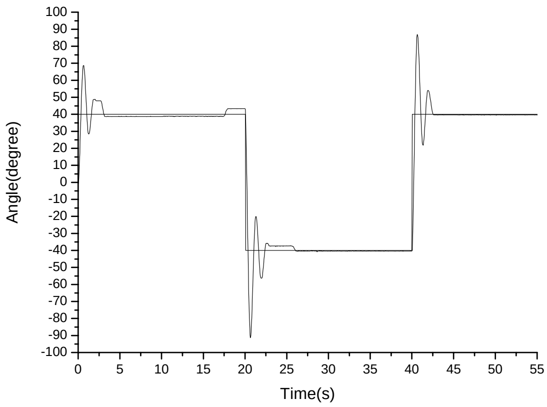


Fig.12 The response of the PI control system with network delay

The control performance of the networked predictive control strategy is given in Figure 13

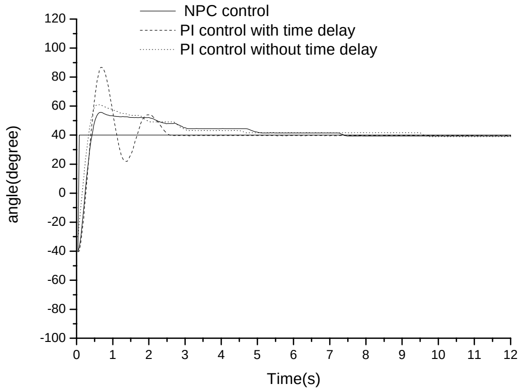


Fig. 13 Comparison of different control strategies for the networked servo control system

Where the same PI controller as that given above was used, and the forward and feedback delay compensators are employed. It is clearly seen from the experiment results that the NPC for a system with network delay has similar control performance to the PI control for the system without network delay. This confirms that the NPC can compensate the network delay effectively.

## 5. CONCLUSIONS

This paper has studied the design and implementation of networked predictive control systems. It has proposed a networked predictive control scheme to address the problem of controlling a system over a network, the paper has presented both off-line and real-time simulation studies, as well as a real-time NPC implementation of laboratory test rigs. It has been shown that the NPC is an active network delay compensation method. Its ability to compensate for the network delay has been demonstrated.

## REFERENCE


F. Lian, J. R. Moyne, and D. M. Tilbury (2001), "Performance evaluation of control networks: Ethernet, controlNet, and deviceNet," *IEEE Contr. Syst. Mag.*, vol. 21, pp. 66 - 83.

L. G. Bushnell (2001), "Guest editorial - Networks and control," *IEEE Contr. Syst. Mag.*, vol. 21, pp. 22 – 23.

Walsh, G.C. and H. Ye (2001), *Scheduling of networked control systems.*. 21(IEEE Contr. Syst. Mag): p. 57-65.

Samsung, *4510B Product Overview*. 2000. p. 1-3.

MathWork, *Real-Time Workshop for Use with simulink*. p. 29-40. 2004

Tham, M.T. (2002), “Internal Model Control, in Introduction to Robust Control”. *Chemical and Processs Engineering*. p. 1-9.

W. K. Ho, T. H. Lee, H. P. Han, and Y. Hong (2001), "Self-tuning IMC-PID control with interval gain and phase margins assignment," *IEEE Trans. Contr. Syst. Technol.*, vol. 9, pp. 535 – 541.

A. T. Bahill (1983), "A simple adaptive Smith-predictor for controlling time-delay systems: A tutorial," *IEEE Contr. Syst. Mag.*, vol. 3, pp. 16 – 22.

G P Liu, J Mu and D Rees (2004) “Networked predictive control of systems with random communication delay”, *Proceedings of the UKACC International Conference on Control*, Bath, UK, ID-015.