REAL-TIME SOFTWARE-IN-THE-LOOP SIMULATION FOR CONTROL EDUCATION

Wook Hyun Kwon, Soo Hee Han, Joon Sok Lee and Hyung Seok Kim

School of Electrical Engineering
Seoul National University, Seoul 151-742, KOREA
FAX:+82-2-871-7010, Tel:+82-2-880-7314
E-mail:whkwon@cisl.snu.ac.kr

Abstract: This paper provides a real-time implementation method for software-in-the-loop (SIL) simulation for control systems, primarily for control education in large classes of big universities where many experiment devices are required to accommodate a lot of students. The implementation is made up of a PC for the controllers, a PC for the plant, an open network, a general-purpose computer-aided control system design (CACSD) package, a model toolbox, and a network box that manages four networks: Ethernet, serial, parallel, and analog to digital/digital to analog (AD/DA) networks. Among various models in usual textbooks, 10 models are animated and can be simulated with other PC for the controllers. The four networks are investigated in terms of control issues such as sampling interval, network-induced time delay, use with multiple I/O points and data synchronization. Specially, to describe the mechanism of Ethernet precisely, the Petri net model is used. A performance evaluation of software-in-the-loop simulation is carried out subject to a computation delay and a sampling interval. To reduce the effects of the time delay, particularly for fast plants, a time-scaling method is introduced for slow and fast motions. A network box is introduced for easy connection and programming of networks. A model toolbox is introduced for the various models of control systems. It is demonstrated that this real-time software-in-the-loop simulation will be very useful, particularly for control education.

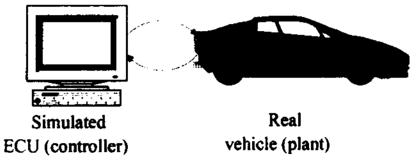
Keywords: Software-in-the-loop simulation; Real time; Education.

1. INTRODUCTION

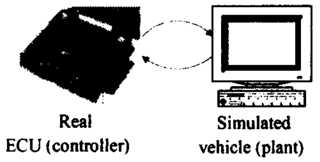
In recent years, algebraic or numerical computations, various plots, and repeated trials are necessary to design and analyze control systems to satisfy given performance specifications. Many computer-aided control system design (CACSD) packages have been developed for the design and analysis of control systems, and have emerged as indispensable tools. They are widely used at universities and research centers for control education and research [1-8].

In real control systems, the controller and plant are separated from each other and exchange a several analog signals for continuous variables and a few digital signals for event variables. However, for simplicity and convenience, CACSD packages are executed in a single personal computer (PC), where a plant and a controller are simulated using off-line simulation. To reflect the more realistic situation, some parts of the real system can be simulated with while connected to the rest of the real system. For example, a real plant or a real controller can be experimented with while connected to a simulated controller or a simulated plant, respectively. The former is called rapid control prototyping (RCP) and the latter is called hardware-in-the-loop (HIL) simulation (Figure 1) [9, 10].

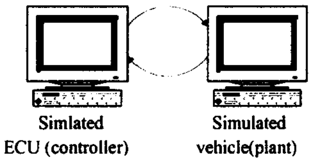
Although these approaches have many advantages, there is also a disadvantage: they are expensive. Both of the RCP and the HIL require a hardware or a device which may be expensive. Specially, for large classes in big universities, many experiment devices are required to accommodate a lot of students. Our approach can provide inexpensive systems in which the simulated process and the simulated controller can be con-



(a) Rapid Control Prototyping



(b) Hardware-In-the-Loop Simulation



(c) Real-time simulation using two PCs

Fig. 1. Real-time simulations (RCP, HIL, and SIL)

nected to each other and run in real-time. We call this method software-in-the-loop (SIL) simulation (Figure 1(c)), because only software exists in the control loop. SIL simulation is particularly useful for control education because it is helpful for students to understand the independent behaviors of various plants and controllers at low cost. HIL simulation has been widely investigated and utilized [9-11], but there have been few investigations of SIL simulation.

This software-in-the-loop simulation should meet the following control-oriented requirements: short sampling intervals should be possible to control relatively fast plants; plants with multiple I/O points should be handled; the input data and the output data between the two parts must be synchronized in order to process the same data at the same time; programming of the control algorithm, the plant model, and their connections must be easy; the cost must be low for wide use, particularly for control education for large classes in big universities.

For real-time SIL simulation, two independent PCs are necessary for low cost and easy use, together with a general-purpose CACSD package for easy programming and the ability to use common networks for multiple I/O points. Nowadays, Ethernet, serial, and parallel ports are standard components for PCs [12, 13], even for notebook computers. PCs in most computer laboratories are all connected with Ethernets. The network with AD/DA can be used at additional cost to reflect real situations [13] since this network is involved in the quantization error and noises coming from transformation to analog or digital signal.

The above control-oriented requirements must be carefully integrated with network-oriented issues such as scheduling, media access control, and media transmission delay. There exist few discussions of these issues, particularly for real-time SIL. Many CACSD packages have been developed [1, 2, 5, 7, 14, 15, 16], but CACSD packages suitable for real-time use of two PCs with open networks are very few. This paper will investigate how to achieve the above requirements such as data synchronization, multiple I/O handling, and data transmission speed for a short sampling interval in four network types. Specially, to describe the mechanism of Ethernet precisely, the Petri net model is used. The four network types usually have different programming and connection. Therefore, for easier programming and connection, a block-oriented network box , a specialized software, will be suggested in this paper. This network box will be implemented and added to an existing CACSD package, CEMTool/SIMTool [17], which was developed at Seoul National University.

A SIL simulation inevitably has computation and network delays for both the controller and the plant. Since fast plants may not be simulated in real-time by a standard SIL simulation, a slow motion function is suggested in this paper to handle fast plants. The slow motion function is achieved using a time-scaling method and its usefulness is analyzed. The dynamics using the time-scaling method will make controller design possible.

For control education, many models will be implemented in a model toolbox so that users can select a model for experiments. The suggested real-time SIL simulation can be carried out in computer laboratories with many PCs or even in classrooms with notebook computers.

This paper is organized as follows. In Section 2, real-time SIL simulation and its requirements are introduced. In Section 3, some plants cited in usual textbooks on automatic control will be introduced and several plants among those plants will be animated and programed with dynamics. The comparison with MATLAB on some peripheral functions will be shown. In Section 4, communication characteristics of four networks are investigated. In Section 5, performance evaluation of real-time SIL simulation is carried out subject to computation and network delays of the plant and the controller. In Section 6, a slow motion function is introduced to get accurate responses for fast plants, and the function is implemented by a time-scaling method. In Section 7, a network box is suggested for easy programming of networks. In

Table 1. Plant list cited in wide-used textbooks

Plant name	A	B	C	D	E
Mass Spring System	O	O	O	O	O
Inverted Pendulum	O	O	O	O	O
Ball Balancer(Ball and Beam)	O		O	O	
Magnetic Ball Levitator	O		O	O	
Digital Tape Transport	O		O	O	
Vehicle Suspension System	O	O	O	O	O
Boiler System			O		
Crane with hanging load			O	O	
Lyquid-Level Control system	O	O	O	O	O
Satellite Attitude System	O	O	O	O	O

Section 8, a model toolbox with the various plant models is introduced. In Section 9, experiment results are shown for a slow plant and a fast plant using the network. Finally, our conclusions follow in Section 10.

2. THE IMPLEMENTED PLANTS AND PERIPHERAL FUNCTION

There are many textbooks on automatic control used in universities. Most of them take real plants for example to explain the concept of control and related mathematics. For given plant dynamics, control can be calculated systematically on the ground of formula provided by textbook. Combining the calculated control and given plant dynamics, off-line simulations are conducted usually. The following table (1) shows the plants cited in several textbook.

The notation of O means that there are corresponding plants in textbooks. A, B, C, D and E mean text books written by Kuo, Ogatta ,Dorf, Franklin and Dutton respectively. The surveyed plants are animated and can be simulated and programed by graphic program. Two plant program among those plants are shown in figure (2) and (3). In the inverted pendulum animation program, the applied force and the inclined angle are shown easily. Additionally, the numerical value are indicated in the upper position of window. In the magnetic ball levitator, the ball moving up and down by electric filed is visualized in addition to numerical value such as position of ball and input voltage. Besides that, all plants listed in table are implemented. In this paper, the water tank system is selected to explain in more detail.

To enhance the efficiency of simulation and provide the convenient user interface, the recent CACSD packages are combined with hardware, executed on real-time OS and visualized using graphical programming. To shorten the calculation time, the C-code replacing macro code are generated and executed on hardware in real time. The real-time OS guarantees the calculation time interval which makes controller work properly.

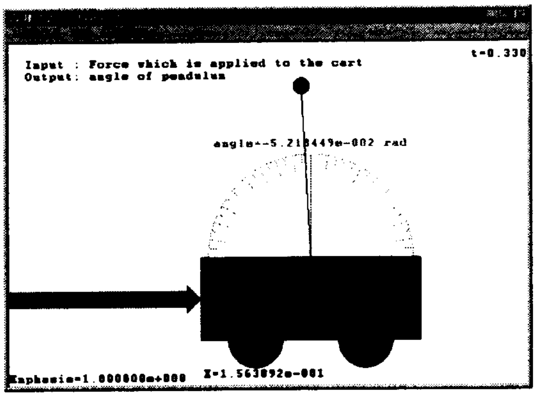


Fig. 2. The animation program for Inverted Pendulum

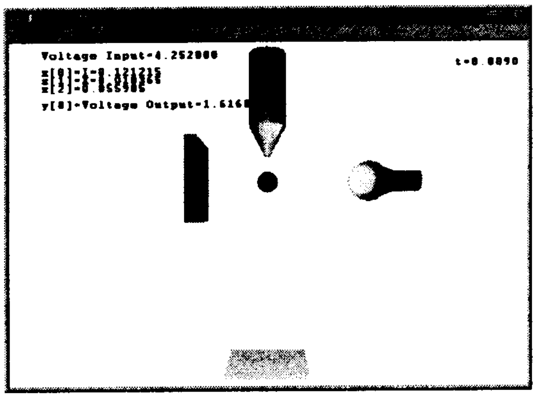


Fig. 3. The animation program for magnetic ball levitator

Table 2. The comparison between the MATLAB and CEMTool

	MATLAB	CEMTool
Support of control Dev. Pro.	HIL, RCP	HIL, RCP, SIL
Code generation	O	O
Execution on real time OS	X	O
Communication using LAN	X	O
Animation	O (with H/W)	O
cost	expensive	medium

The following table (2) shows the implemented function of two CACSD package. Spcially, the communication using LAN in CEMTool provides the simulation environments without much expense. In CEMTool Animation programs are provided in only software form, not based on hardware.

In real class, the following guide is recommended. Two PCs are assigned to one team composed of about two or three students. Several plant models and the design specifications for the corresponding plants are provided to each team. Then, students will design controllers satisfying the design specifications. For demonstration, instructors can use two notebook PCs in the class. We found that most students prefer this design approaches using two PCs to conventional approaches using one PC.

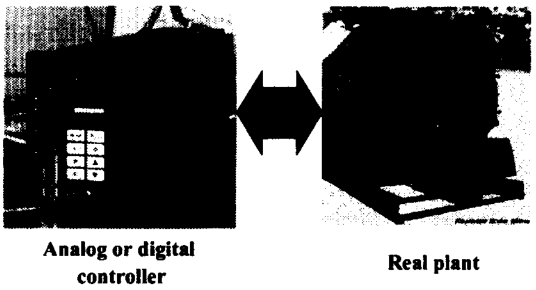


Fig. 4. Real control system

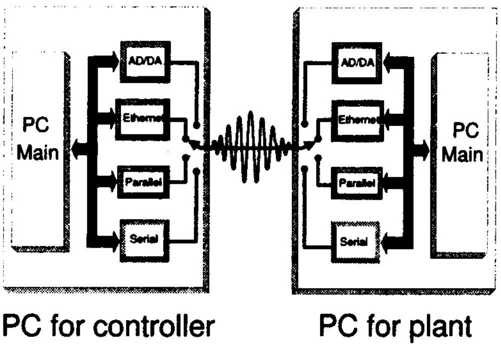


Fig. 5. Real-time SIL simulation system using two PCs and a network

3. REAL-TIME SOFTWARE-IN-THE-LOOP (SIL) SIMULATION AND REQUIREMENTS

Consider a real control system, Figure 4, in which the left side shows a loop controller that is very widely used in industry as the core device of controllers, and the right side shows a boiler plant. The separated controller and plant can be modeled by two separated PCs (Figure 5) with an open network. In the plant PC, we can model many processes, such as a water tank, a two-dimensional moving system, a spring-mass system, a boiler, or an inverted pendulum. In the control PC, we can introduce several control algorithms, such as PID, LQG, fuzzy control, or pole assignment.

For this real-time SIL simulation, the network and the computers must meet the following control-oriented requirements. 1) The data from the controller PC and the plant PC must be synchronized for control purposes. That is, data for a given time must be processed together and should not be mixed with other data from a different time. 2) Network-induced delays and the computation of control algorithm and plant dynamics must be short in order to achieve short sampling intervals. 3) In each sampling interval, data transmission and the computation of control algorithms in the controller PC must be completed as well as data transmission and the computation of plant dynamics in the plant PC. 4) The network must be able to handle multiple I/O points for multivariable plants. 5) The network must be reliable in order to send data without noise.

It is not easy to program different networks for such a system. Therefore, a network box is necessary to simplify programming for control engineers. A network box has the following software requirements. 1) It must include all necessary communication interfaces. 2) It must be easy to use. 3) It must be able to modify various network parameters such as I/O addresses. 4) It must handle the above control-oriented requirements. 5) It must be combined with a general purpose CACSD package.

In the next section, characteristics of the necessary networks and their implementation are investigated for real-time SIL simulation.

4. TIME DELAY AND SAMPLING TIME IN OPEN NETWORKS

Sampling interval, data synchronization, and network delay with multiple channels will be investigated for each network for real-time SIL simulation.

A controller repeats the sequence of receiving control-input data, computing the control algorithm, and sending control-output data. The period of these processes is called a sampling interval. A sampling interval is determined by the computation time of the control algorithm and the network time delay. Once the sampling interval is determined, a real-time clock in the PC that can be programmed to produce hardware interrupts at desired time intervals guarantees the sampling interval in real-time SIL simulation. This real-time clock generates clock-ticks, with an interval of 18.2Hz in this paper.

Given the condition that the sampling interval of the plant PC and the controller PC must be the same, the sampling interval (t_{sin}) must be set by the following condition for all networks:

$$t_{sin} \geq max(t_{ct} + t_{con}, t_{ct} + t_{pl})$$

where t_{ct} is the communication delay time, t_{con} is the computation time for the controller, and t_{pl} is the computation time for the plant.

Ethernet, serial, and parallel ports are standard components for most PCs, including notebook computers. Among these, Ethernet is fast, relatively noiseless, and has a long communication distance. AD/DA is closer to the real situation, but is more expensive to purchase.

1) Ethernet network: Ethernet is a type of medium access control protocol. The topology of an Ethernet network is based on random access of the physical medium. An Ethernet network has three kinds of networks such as user-defined protocol, UDP (user datagram protocol) /IP (internet protocol), and TCP (transmission control protocol) /IP. An Ethernet network with TCP/IP comprises a set of communication protocols that is widely used nowadays [18-20]. TCP/IP can control transmission errors and is suitable for networks having a high probability of transmission errors. However, since the control system in this paper is deterministic and the collisions do not occur in the network, TCP/IP can cause unnecessary transmission delays because of transmission error checks. On the other hand, when we use only IEEE 802.3 CSMA/CDAU [21], we need to implement different device drivers for different systems. It is noted that the Ethernet network with UDP/IP, which has good portability though it cannot check transmission errors, is one way to get good performance in real-time SIL simulation.

The method used for synchronization of the plant PC and the controller PC is that one PC waits for input data until the other PC writes its output data at the Ethernet buffer. The two PCs are synchronized by repeating the data-writing, the data-waiting, and the data-reading. Therefore, only one real-time clock is needed for the sampling interval of the two PCs because one PC is synchronized automatically to the other PC by this repetition.

Figure 6 shows the data transmission cycle through the Ethernet network. The procedure for this is as follows:

Step 1. At the start of the sampling interval of the plant PC, the plant PC sends the plant-output data (PD0) to its own send buffer and then via the network to the receive buffer of the controller PC. This process is initiated by the "send" command of the communication program.

Step 2. The control-input data (PD0) is achieved by the "receive" command. If the control-input data does not exist in the receive buffer, the controller PC waits for the control-input data until the control-input data exists in the receive buffer. The plant PC achieves the control-output data.

Step 3. Each PC computes either the control algorithm or the plant dynamics.

Step 4. The plant PC waits for the next sampling interrupt. The controller PC sends the control-output data (CD1).

Step 5. At the start of the next sampling interval, all processes (*Step 1 - Step 4)* are repeated.

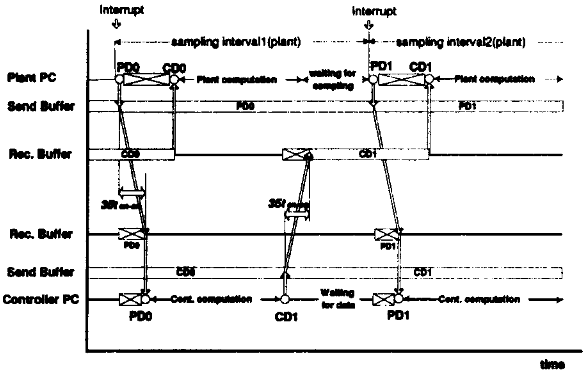


Fig. 6. Ethernet network timing chart for one channel

The transmission rate of Ethernet is very fast in real-time SIL simulation. The ideal Ethernet network transmission time is as fast as $10M$ bit per sec. This means that Ethernet does not seem to have the disadvantage of performance degradation depending on the number of I/O points. For multiple I/O points, the data transmission is accomplished by sending one packet with as many data values as there are I/O points as well as data for the UDP/IP protocol. If the plant has n I/O points, the total Ethernet communication delay time in each sampling interval is represented as

$$t_{ct} \cong 2 * t_{en-en} * (2 * n + 33)$$

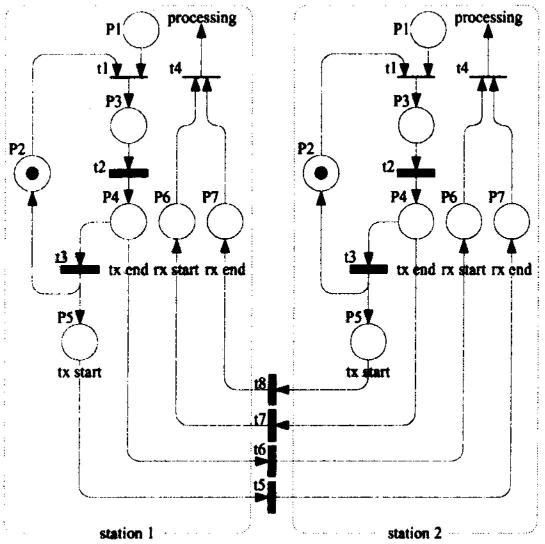
where t_{en-en}, the one-byte data transmission time between two Ethernet stations, is calculated using the baud rate of the Ethernet network and the additional 33 bytes are needed for the UDP/IP protocol.

Real-time SIL simulation with this network can be used widely at no extra cost.

In figure (7), the Petri net model for Ethernet is shown. Some delay factors such as transmission delay and propagation delay is described. The one-byte data transmission time between two Ethernet stations (t_{en-en}) is calculated using t_2, t_3 and t_5.

2) AD/DA network: An AD/DA network transmits analog data between the PCs. Real-time clocks guarantee the sampling intervals in the plant PC and the controller PC. For an AD/DA network, data synchronization is hard to implement because neither a handshaking method nor any other methods for the data synchronization can be used. Therefore, start times for the plant and the controller may be different. Though the start times are different, the analog data is always latched in the AD/DA board so that both PCs can always accept the data.

For multiple I/O points, the AD/DA network needs multiple AD converters and DA converters. The cost for multiple I/O points is therefore high. In contrast to other networks, the AD/DA network is exposed to noise that contributes to performance deviation.



P1: transmission queue
P2: transmission ready state
P3: transmission buffer
t2: transmission delay
t3: inter frame time
t5~t8: propagation delay

Fig. 7. Petri net model of Ethernet network between two PCs

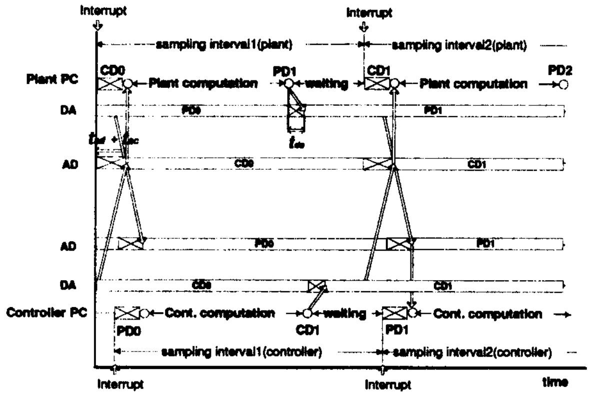


Fig. 8. AD/DA network timing chart for one channel

Figure 8 shows the data transmission cycle through the AD/DA network. The procedure is as follows:

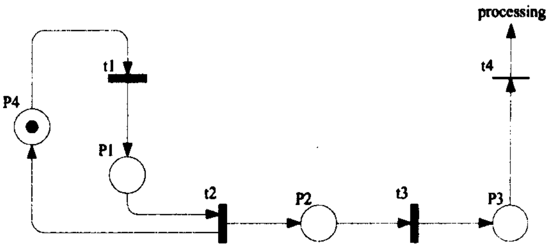
Step 1. At the start of the sampling interval for the plant, the plant PC accepts the plant-input data (CD0) using the AD converter. At the start of the sampling interval for the controller, the controller PC accepts the control-input data (PD0) using its own AD converter.

Step 2. Each PC uses its input data to compute the control algorithm and the plant dynamics.

Step 3. The data for control (CD1) and plant (PD1) outputs are converted into analog data by each DA converter.

Step 4. At the start of the next sampling interval, all processes (*Step 1 - Step 3)* are repeated.

If the plant has n I/O points, the total AD/DA communication delay time (t_{ct}) in each sampling interval is represented as



P1: in D/A conversion
P2: in transmission
P3: in A/D conversion
P4: in suspending
t1: sampling time
t2: D/A conversion time
t3: propagation delay
t4: A/D conversion time

Fig. 9. Petri net model of AD/DA network between two PCs

$$t_{ct} \cong (t_{ad} + t_{da} + t_{ac}) * n$$

where t_{ad} is the analog to digital conversion time, t_{da} is the digital to analog conversion time, and t_{ac} is the channel acquisition time.

In figure (9), the Petri net model for AD/DA network is shown. t_4 of figure (9) means the time spending to transform analog signal to digital signal. This time is the main factor for delay time between two PCs.

3) Serial network: A serial network is frequently used for connecting two PCs. A serial network has the advantage that the network cost is low and the disadvantage that the data transmission time is longer than for other networks while the distance of the data transmission is limited. In real-time SIL simulation, it is a good communication network for a slow plant, a short distance between PCs, and a small number of channels for the plant. The procedure for transmission in the serial network is similar to that for the Ethernet network. For multiple I/O points, the data transmission is accomplished by sending the same number of data values as there are I/O points between the transmission start data and stop data.

If the plant has n I/O points, the total serial communication delay time in each sampling interval is represented as

$$t_{ct} \cong 2 * t_{si-si} * (2 * n + 2)$$

where t_{si-si} is the one-byte data transmission time in the serial network, and is calculated using the baud rate of the serial network. The additional two bytes are the start and the stop data for data identification.

4) Parallel network: For a parallel network, a printer port (parallel port) of a PC can be used. In a parallel port of a PC, receive buffers, re-

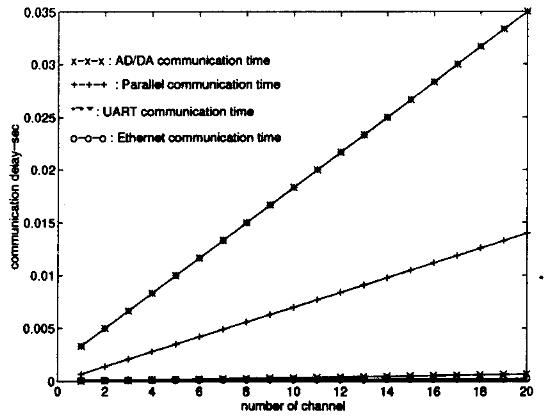


Fig. 10. Communication time for four networks by number of channels

ceive lines, send buffers and send lines are not provided and only a two-way receive-send line is provided. Because of this characteristic, parallel communication using a hand-shaking method is designed. Synchronization between the plant and the controller occurs automatically because of the hand-shaking method. Therefore, only one real-time clock is needed for the sampling interval of the two PCs. For multiple I/O points, the parallel network needs no additional hardware. The data transmission time is longer in proportion to the number of I/O points.

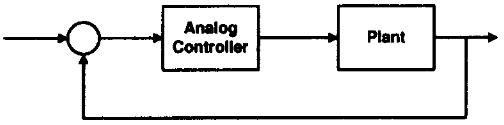
If the plant has n I/O points, the total parallel communication delay time in each sampling interval is represented as

$$t_{ct} \cong 7 * t_{pw} * n$$

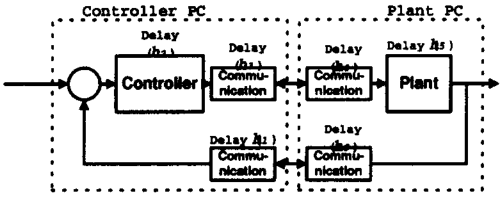
where t_{pw} is the ready time for acknowledgment or data reception.

5) Comparison: In Figure 10, the delay times per channel of the four networks are shown, assuming that the parameter values are as follows: $t_{ad} = 50$ μ sec, $t_{da} = 25$ μ sec, $t_{ac} = 5$ μ sec, $t_{si-si} = 0.417$ m sec, $t_{en-en} = 1.6$ μ sec, and $t_{pw} = 100$ μ sec. In terms of data transmission speed, the Ethernet network is the fastest of the four networks.

A comparison of the four networks is shown in Table 1. We can see that the Ethernet network may be better than the others for real-time SIL simulation considering several requirements. In particular, it is better in terms of time delay, communication distance, reliability, and cost. The AD/DA network is good only because it is closer to the real variable data.



(a) Ideal delay-free controller and plant



(b) SIL simulation
with both plant and controller delays

Fig. 11. Control system classification according to delay

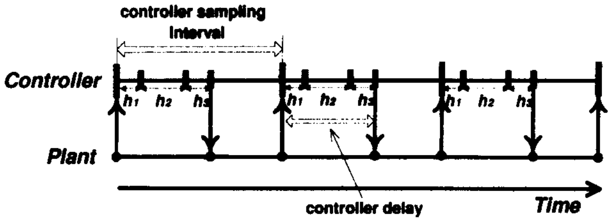


Fig. 12. Total delay in RCP (Figure 11(b))

5. TOTAL DELAY TIME IN-THE-LOOP AND PERFORMANCE EVALUATION

In an RCP, simulated digital controllers with computers have computation and communication delays [22-25], while plants do not have such delays. However, in a real-time SIL simulation, both the plant PC and the controller PC have computation and communication delays because of the computation of dynamics and the communication of the input and the output data. These delays can make the performance of the controlled system worse or even make it unstable [25, 26]. Hence, it is necessary to evaluate the effect of delays in these systems.

The ideal delay-free analog controller and delay-free plant are depicted in Figure 11(a). Compared with this, two different cases can be considered: a delay-free plant with a controller delay in an RCP (Figure 11(b)) and both plant and controller delay in a SIL simulation (Figure 11(c)). The performance of the two cases is compared with that of the ideal delay-free analog system.

The delay in an RCP (Figure 11(b)) is modeled in Figure 12, which shows the controller delay [22]. As we can see in Figure 12, the controller has a data-input delay (h_1), a computation delay for the control algorithm (h_2), and a data-output delay (h_3). However, the plant does not have any delays.

The delay in a real-time SIL simulation, especially for the Ethernet network case, is modeled in Figure 13, which shows the controller delay and

Interface	Ethernet	AD/DA	Serial	Parallel
Fast plant experiment	good	good	bad	medium
Slow plant experiment	good	good	good	good
Time delay	short	medium	long	medium
Comm. time for multiple I/O points	fast	fast	slow	medium
Communication distance	long	short	short	medium
Data reliability	good	bad	good	good
Reflection of real plant data	bad	good	bad	bad
Interfacing price	medium	expensive	cheap	cheap

Table 3. Evaluation of four networks as education tools

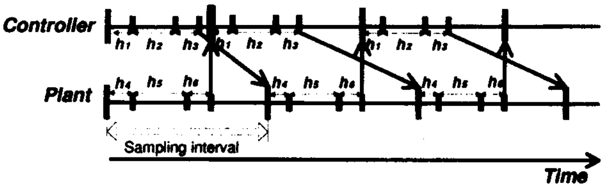


Fig. 13. Total delay in SIL simulation (Ethernet network, Figure 11 (c))

the plant delay. As we can see in Figure 13, the Ethernet network always has one sampling delay at the control input and one sampling delay at the plant input. The serial and parallel networks have the same delay as the Ethernet network. However, the total delay time of the AD/DA network varies from 1 to 3 sampling delays according to the computation delays, the network-induced delays, and the starting times of the sampling intervals in the two PCs.

For the performance evaluation, a water tank system is selected as the plant. The dynamics of the water tank system are as follows:

$$\dot{h}_1 = -\frac{1}{R_1C_1}h_1 + \frac{1}{R_1C_1}h_2 + \frac{1}{C_1}W$$
$$\dot{h}_2 = \frac{1}{R_1C_2}h_1 - (\frac{1}{R_1C_2} + \frac{1}{R_2C_2})h_2ber \quad (1)$$

where h_1 and h_2 are the levels of the water tanks, R_1 and R_2 are the resistances of valves, C_1 and C_2 are the capacitances of the water tanks, and W is the amount of water flowing from the inlet tap. In this system, the control objective is that the level of the second water tank is regulated at the desired value. In the following simulations, the specifications of the water tanks are that $h_1 = 100m$, $h_2 = 100m$, $R_1 = 0.009$ sec$/m^2$, $C_1 = 10m^2$, $R_2 = 0.2$ sec$/m^2$, and $C_2 = 3m^2$. The time constant of this system is 0.0402. This describes a rather fast water tank system. A PID controller is considered as a controller for the water tank system. In the following simulations, PID coefficients are $K_p = 100$, $K_i = 40$, and $K_d = 5$.

To judge the effect of the delay in an RCP, step responses of a closed-loop water tank system are shown with respect to the time delay in Figure 14.

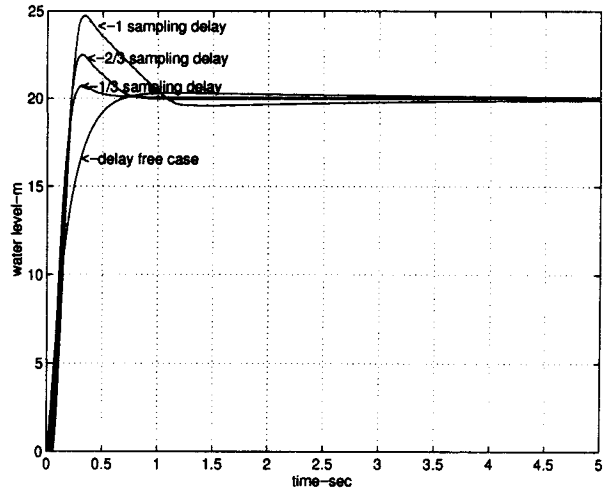


Fig. 14. Effect of delay in RCP

In this figure, "x sampling delay" means that the time consumption for the computation of control algorithm and communication is x times the sampling interval. In this simulation, the sampling interval of the controller is 0.055sec. As the delay of dynamics computation and data communication grows larger, the deviation from the ideal delay-free analog system grows larger.

To judge the effect of delay in a real-time SIL simulation, step responses of a closed-loop water tank system with respect to the total delay are shown in Figure 15. In this simulation, the sampling interval of both the plant and the controller is 0.055sec. We can see that in a real-time SIL simulation the deviation from the ideal system is much more than in the RCP, because there is more delay in the real-time SIL simulation than in the RCP.

6. TIME-SCALING METHOD FOR SLOW MOTION FUNCTION

In real-time SIL simulation, if the communication delay and the dynamic computation are substantial, the simulation results can be quite different from the original system, as can be seen in the previous section. In real-time SIL simulation, delay is unavoidable because of the dynamics computations and the network delay. We need a

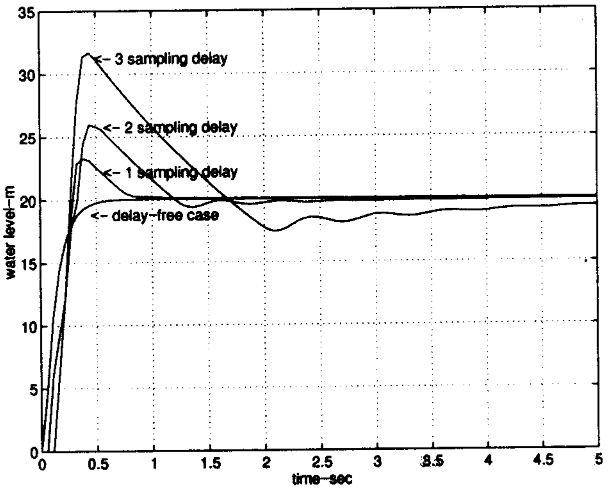


Fig. 15. Effect of delay in a SIL simulation

novel method so that this control system can deal with fast real plants and controllers. We suggest a time-scaling method to handle this problem.

We observe that if the plant and the controller have slow dynamics, the effect of the delay is reduced relatively. Thus, if a plant is fast compared to the dynamics computation and network delay in real-time SIL simulation, we would rather slow down the plant and controller dynamics intentionally and see the dynamics accurately in slow motion.

Let an original system be defined as follows:

$$\frac{dx(t)}{dt} = f(x(t), u(t))$$
$$y(t) = g(x(t)).$$

A new time variable($t' = \alpha t$) is introduced. Then the original system becomes

$$\frac{dx(\frac{1}{\alpha}t')}{dt'} = \frac{1}{\alpha} f(x(\frac{1}{\alpha}t'), u(\frac{1}{\alpha}t'))$$
$$y(\frac{1}{\alpha}t') = g(x(\frac{1}{\alpha}t')).$$

For a controller such as

$$u(t) = u(r(t), y(t)),$$

the time-scaled controller equation becomes

$$u(\frac{1}{\alpha}t') = u(r(\frac{1}{\alpha}t'), y(\frac{1}{\alpha}t')).$$

It is noted that the reference $r(t)$ must be also scaled as $r(\frac{1}{\alpha}t')$.

From this time-scaled system, the time-scaled response ($\alpha \neq 1$) is obtained. To obtain the real response of the original system, the time-scaled response is again scaled as $t = \frac{1}{\alpha}t'$. For verification, simulations for control systems with the following conditions are performed:

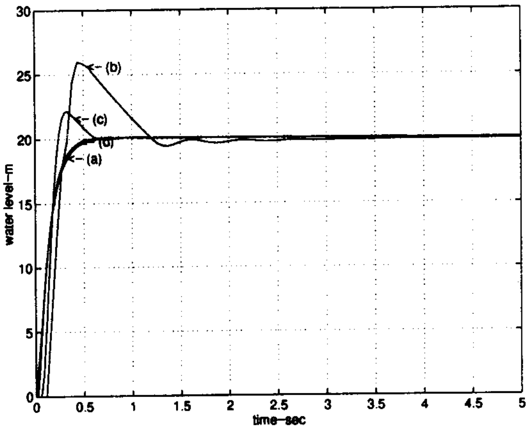


Fig. 16. Comparison of performance

(a) Delay-free controller and delay-free plant.

(b) Real-time SIL simulation: plant and controller delays

(Plant delay = 0.055 sec, Controller delay = 0.055 sec).

(c) Time-scaling method: $\alpha = 2$ in condition (b).

(d) Time-scaling method: $\alpha = 10$ in condition (b).

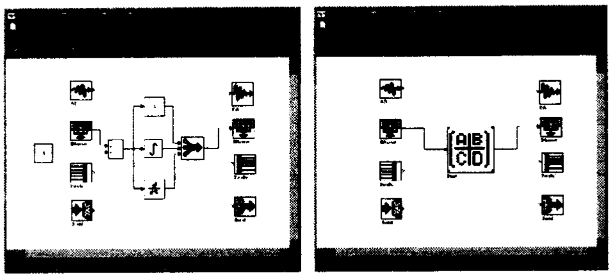
In all these simulations, the controller and plant parameters, and the sampling interval are the same as those of Section 5.

Figure 16 shows step responses for control systems (a), (b), (c), and (d). It is noted that the response times in (c) and (d) are again scaled as $t = \frac{1}{2}t'$ and $t = \frac{1}{10}t'$. The deviation from the ideal system is large in (b) compared to (a). However, in a control system with a large time-scaling coefficient, as in (d), the deviation from the ideal system is much less than that of (b). As the time-scaling coefficient increases, the response of the time-scaled system approaches that of the ideal system more closely.

7. A SUGGESTED NETWORK BOX

In a real-time SIL simulation, it should be easy to program the network between a controller PC and a plant PC. To allow this, we introduce four types of eight network blocks to CEMTool/SIMTool, an object-oriented [27, 28] block diagram graphic editor [29-32], compiler and executor [33], with which a user can implement both plants and controllers. Figure 17 shows a controller and a plant with network blocks, which are programmed using CEMTool/SIMTool.

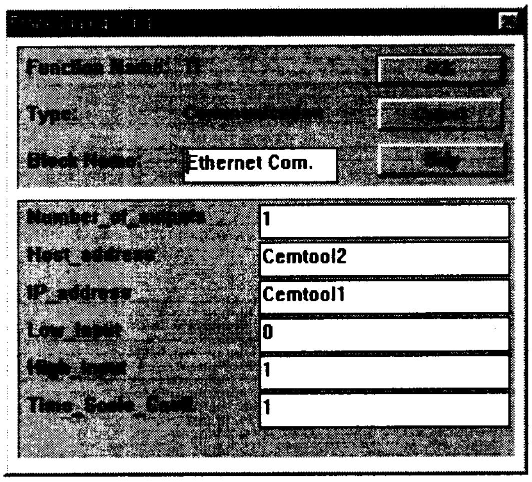
These blocks enable users to specify the parameters needed for communication: the Ethernet network can be programmed with TI/TO blocks with



(a) Controller

(b) Plant

Fig. 17. Controller and plant modeled with network blocks



Block Name: Ethernet Com.

Number_of_outputs: 1

Host_address: Cemtool2

IP_address: Cemtool1

Low_input: 0

High_input: 1

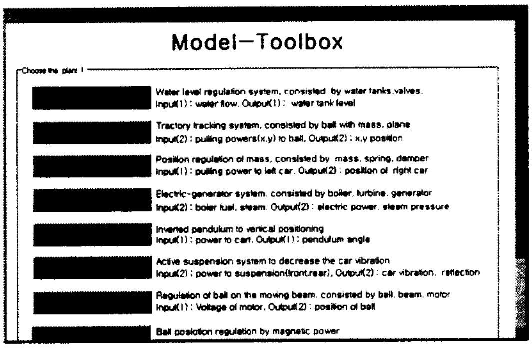
Time_Scale_Coeff: 1

Fig. 18. Input tuning parameters for an Ethernet network

six parameters such as host name, peer name, channel number, high input, low input, and time-scaling coefficient; for the serial network, SI/SO blocks have six parameters such as channel number, COM port, baud rate, high input, low input, and time-scaling coefficient; for parallel network, PI/PO blocks have five parameters such as channel number, LPT port, high input, low input, and time-scaling coefficient; for AD/DA networks, AI/AO blocks have five parameters such as base I/O address, channel number, high input, low input, and time-scaling coefficient. Parameter setting boxes are introduced for each network. For example, one parameter setting box for the Ethernet network input block is shown in Figure 18.

8. A SUGGESTED MODEL TOOLBOX

For the various models, a model toolbox is suggested. This has more than twenty models including a water tank, a two dimensional moving system, a spring-mass system, a boiler, an inverted pendulum, a half car, a ball and beam, a magnetic levitation experiment, a satellite, and a digital positioning system. A part of the model toolbox window is shown in Figure 19. A model can be chosen by a mouse-click in the model toolbox window. For various experiments, it is suggested that the user choose an animation, model parameters, and I/O networks. It is designed so that the specifications of a model can be changed easily using only mouse-clicks and data entry.



Model–Toolbox

Choose the plant !

Water level regulation system, consisted by water tanks,valves.
Input(1) : water flow, Output(1) : water tank level

Tractory tracking system, consisted by ball with mass, plane
Input(2) : pulling powers(x,y) to ball, Output(2) : x,y position

Position regulation of mass, consisted by mass, spring, damper
Input(1) : pulling power to left car, Output(2) : position of right car

Electric-generator system, consisted by boiler, turbine, generator
Input(2) : boiler fuel, steam, Output(2) : electric power, steam pressure

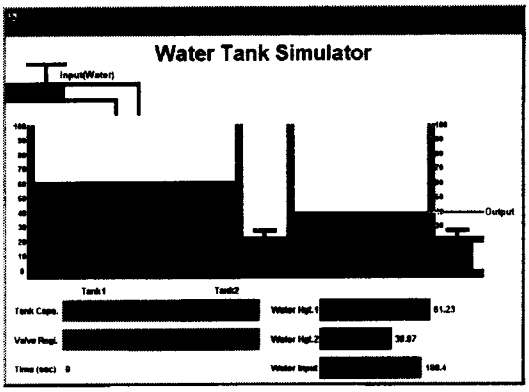
Inverted pendulum to vertical positioning
Input(1) : power to cart, Output(1) : pendulum angle

Active suspension system to decrease the car vibration
Input(2) : power to suspension(front,rear), Output(2) : car vibration, reflection

Regulation of ball on the moving beam, consisted by ball, beam, motor
Input(1) : Voltage of motor, Output(2) : position of ball

Ball posiotion regulation by magnetic power

Fig. 19. A part of the model toolbox window



Time (sec) 0

Water Hgt.1 61.23

Water Hgt.2 38.87

Water Input 198.4

Fig. 20. Water tank: animation part

Among these models, the water tank system is selected for the real-time SIL simulation in the next section. The animation part of the water tank system is shown in Figure 20.

9. REAL-TIME SIL SIMULATION

In this section, a real-time SIL simulation is performed for both slow and fast water tank systems.

The PID controller designed for the water tank system is shown in Figure 21. It is programmed with CEMTool/SIMTool in the controller PC, which has a gain block, an integral block, a derivative block, a plot block, and other blocks. The reference level of the right water tank can be set in CEMTool/SIMTool, in this experiment as 20 m. In the following experiments, coefficients for the PID controller are $K_p = 100$, $K_i = 40$, and $K_d = 5$. The sampling interval of the controller PC is 0.055 sec, which is set as one clock-tick time in the controller PC. The plant PC is synchronized with this sampling interval.

The controller PC communicates with the plant PC by selecting one of the four different networks,

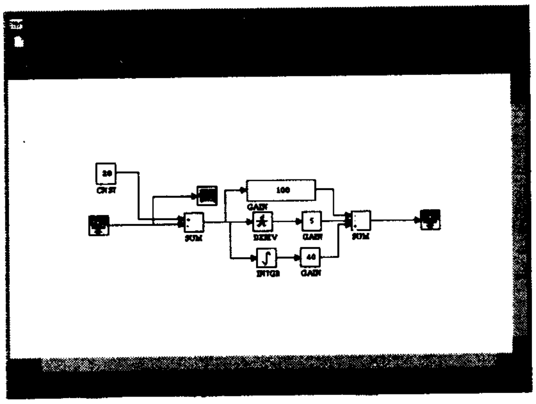


Fig. 21. SIMTool blocks for a PID controller

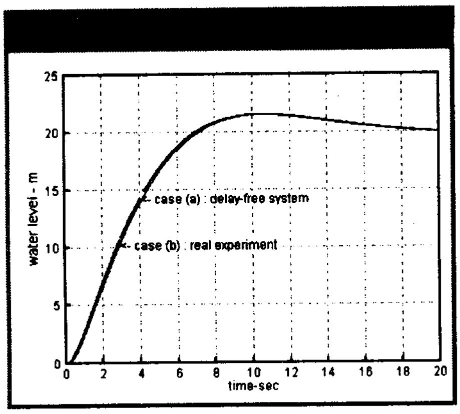


Fig. 22. Real-time SIL simulation for a slow system

as in Figure 17. For the following experiments, Ethernet communication between the controller PC and the plant PC is used.

Figure 22 shows the step response for a slow water tank system. In this experiment, the specifications of the water tanks are that $h_1 = 100m$, $h_2 = 100m$, $R_1 = 0.01\text{sec}/m^2$, $C_1 = 200m^2$, $R_2 = 0.01\text{sec}/m^2$, and $C_2 = 100m^2$. The time constant of this system is 0.8. Case (a) in Figure 22 shows the response of the delay-free system, which is computed in a single PC without a network delay. Case (b) in Figure 22 shows the response of the real-time SIL simulation. We can see that the real-time SIL simulation gives an accurate response, comparable to the delay-free system.

Figure 23 shows a step response for a fast water tank system. In this SIL simulation, the specifications of the water tanks are $h_1 = 100m$, $h_2 = 100m$, $R_1 = 0.0009\text{sec}/m^2$, $C_1 = 10m^2$, $R_2 = 0.2\text{sec}/m^2$, and $C_2 = 3m^2$. The time constant of this system is 0.0402, which is smaller than the sampling interval 0.055sec. Case (a) in Figure 23 shows the response of the delay-free system. Case (b) in Figure 23 shows the response of the real-time SIL simulation. We can see that the real-time SIL simulation is poor since the delay time is relatively large for the fast plant. This problem can be handled by the time-scaling method. Case (c) in Figure 23 shows the response with time-scaling ($\alpha = 10$), which is closer to case (a). This shows that by introducing the time-scaling method real-time SIL simulation can give accurate responses even for fast plants.

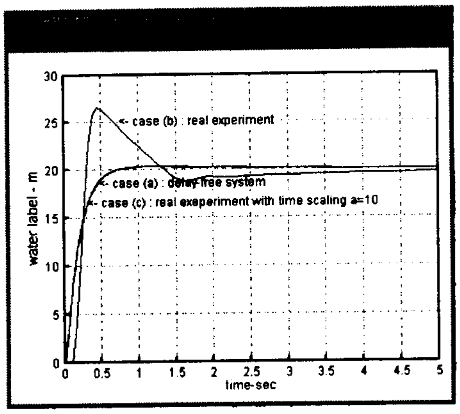


Fig. 23. Real-time SIL simulation for a fast system

Accurate experiments even for fast systems by use of the time-scaling method is a prominent advantage of real-time SIL simulation.

10. CONCLUSION

In this paper, in order to experiment with real control systems at low cost, a real-time SIL simulation is suggested, which is composed of two PCs with an open network, a general-purpose CACSD package, a network box, and a model toolbox. Real-time SIL simulation is investigated in terms of data synchronization, network delay, number of I/O points, and sampling interval.

For communication between the controller PC and the plant PC, four networks, Ethernet, serial, parallel, and AD/DA, can be used in the real-time SIL simulation. They are standard components for most PCs including notebook computers. Among these, Ethernet is fast and relatively noiseless, has long communication distance and is common in computer laboratories. Real-time SIL simulation with Ethernet is investigated in detail in this paper and can be used widely at no extra cost. AD/DA is closer to the real situation, but is more expensive. In order to support the four networks, network blocks are implemented in CEMTool/SIMTool.

The effects of computation and communication delays are shown for real-time SIL simulation with both plant and controller delays. A time-scaling method is introduced for real-time SIL simulation for fast plants.

The real-time SIL simulation of this paper has several advantages. It may be very useful for experiments for control education. In particular, it can be demonstrated in classes with only two notebook computers. It is very educational to observe in real-time how various plants behave according to different control algorithms. It is also possible to experiment easily with many different plants for control education since many different plants exist in a model toolbox. After slight modifications, this system could also be used for other areas, including signal processing.

11. REFERENCES

[1] C. J. Herget, "Survey of existing computer-aided design in control systems packages in the united states of America," *IFAC Computer Aided Design in Control Systems*, Beijing, PRC, pp. 59-64, 1988.

[2] Ad van den Boom, "CADCS Developments in Europe," *IFAC Computer Aided Design in Control Systems*, Beijing, PRC, pp. 65-74, 1988.

[3] M. Rimvall, "Computer-Aided Control System Design," *IEEE Control Systems Megazine*, pp. 14-16, Apr., 1993.

[4] Mats Andersson, Dag Bruck, Sven Erik Mattsson, and Tomas Schonthal, "OmSim-An Intergrated Interactive Environment for Object-Oriented Modeling and Simulation," *IEEE/IFAC Joint Symposium on CACSD*, Tucson, Arizona, pp. 285-290, 1994.

[5] Babatunde, A. Ogunnaike, "The Role of CACSD in Contemporary Industrial Process Control ," *IEEE Control Systems Magazine*, 15(2) pp. 41-47, 1995

[6] Rudiger Rutz, Jobst Richert,"CAMeL: An Open CACSD Environment," *IEEE Control Systems Magazine*, 15(2), pp. 26-33 , 1995

[7] John James , Francois Cellier , Grantham Pang , John Gray , and Sven Erik Mattsson, "The State of Computer-Aided Control System Design(CACSD)," *IEEE Control Systems Magazine*, 15(2), pp. 6-7, 1995

[8] Safonov-MG, Chiang-RY, "Cacsd Using the State-Space L-Infinity Theory-A Design Example ," *IEEE Transactions on Automatic Control*, 33(5), pp. 477-479, 1988.

[9] R. Isermann, S. Sinsel, and J. Schaffnit"Hardware-in-the-loop simulation of Diesel-engins for the development of engine control of Diesel-engines for the development of engine control systems," *4th IFAC Workshop on AARTC'97*, pp. 90-92, 1997.

[10] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *5th IFAC Workshop on AARTC'98*, pp. 1-10, 1998.

[11] A. Kimura, and I. Meda, "Development of engine control system using real time simulator," *IEEE Symposium on CACSD*, Dearborn, pp. 157-163, 1996.

[12] D. M. Auslander and C. H. Tham, *Real-time Software for Control*, Prentice-Hall, 1990.

[13] M. F. Hordeski, *Control System Interfaces Design and Implementation using Personal Computers*, Prentice-Hall, 1992.

[14] Pang, Grantham K. H. "Issues in the development of the interactive CACSD package SF-PACK," *American Control Conference*, pp. 1002-1007, 1991

[15] J. Multisilta and S. Pohjolaninen,"Hypermedia in CACSD Education for Distributed Parameter Systems," *IEEE Symposium on CACSD*, Napa, California, pp. 250-254, 1992.

[16] Matt Wette, "Casey: A Computer-Aided Engineering System," *IEEE Symposium on CACSD*, Napa, California, pp. 232-237, 1992.

[17] W. H. Kwon, *Practical Automatic Control using CEMTool*, Cheng Moon Gock, 1996.

[18] Ray, A., "Performance Evaluation of Medium Access Control Protocols for Distributed Digital Avionics," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 370-377, December 1987.

[19] Halevi, Y., and Ray, A., "Integrated Communication and Control Systems: Part I-Analysis," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 367-373, Dec. 1988.

[20] Ray, A., and Halevi, Y., "Integrated Communication and Control Systems: Part II- Design Considerations," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 374-381, Dec. 1988.

[21] IEEE, "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications," 1985.

[22] K. J. Astrom and B. Wittenmark, *Computer-Controlled Systems*, Prentice-Hall, 1990.

[23] K. G. Shin and X. Cui, "Computing Time Delay and Its Effects on Real-Time Control Systems," *IEEE Transactions on Control Systems Technology*, 3(2), pp. 218-224, 1995

[24] S. H. Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems," *IEEE Tr. on CST*, 3(2), June, 1995.

[25] Y. H. Kim , W. H. Kwon , and H. S. Park, "Stability and a Scheduling Method for Network-based Control Systems," *IECON*, pp. 934-939 , 1996

[26] Krtolica, U. Ozguner, H. Chan, H. Goktas, J. Winkelman, and M. Liubakka, "Stability of linear feedback system with random com-

munication delays," *International Journal of Control*, 59(4), pp. 925-953, 1994.

[27] Jobling-CP, Grant-PW Barker-HA, Townsend-P, "Object-Oriented Programming in Control-System Design - A Survey," *Automatica*, 30(8), pp. 1221-1261, 1994.

[28] R. Kroeger, "Using Object-Oriented Programming Standards for Developing Distributed Real-time Control Applications," *3rd IFAC/IFIP workshop on AARTC'95*, Ostend-Belgium, pp. 135-153, 1995.

[29] M. Rimvall, "Interactive environments for CACSD software," *IFAC Computer Aided Design in Control Systems*, Beijing, PRC, pp. 17-26, 1988.

[30] H. A. Barker, M. Chen, P. W. Grant, C. P. Jobling and P. Townsend, "A Man-Machine Interface for Computer-aided Design and Simulation of Control Systems," *Automatica*, 25(2), pp. 311-316, 1989.

[31] D. Xue and D.P.Atherton, "BLOCKM-A Block Diagram Modeling Interface and its Applications," *IEEE Symposium on CACSD*, Napa, California, pp. 242-249, 1992.

[32] O. Ravn, M.Szymkat, "Requirements for user interaction support in future CACE environments," *IEEE/IFAC Joint Symposium on CACSD*, Tucson, Arizona, pp. 381-386, 1994.

[33] Masanobu Koga, Mitsuji Sampei, and Katsuhisa Furuta, "A Compiler of MATLAB to MATX Compiling and Linking of m-files to an Executable Program," *IEEE/IFAC Joint Symposium on CACSD*, Tucson, Arizona, pp. 381-386, 1994.