International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **11**, November **2011**

 ISSN 1349-4198
pp. **6369–6382**

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

Soohee Han[1], Seong-Gyu Choi[2] and Wook Hyun Kwon[2]

[1]Department of Electrical Engineering
Konkuk University
Hwayang-dong, Gwangjin-gu, Seoul 143-701, Korea
shhan@konkuk.ac.kr

[2]ERC for Advanced Control and Instrumentation
School of Electrical Engineering
Seoul National University
Gwanak-ro, Gwanak-gu, Seoul 151-742, Korea
{ csg; whkwon }@cisl.snu.ac.kr

Received May 2010; revised September 2010

Abstract. *This paper suggests a real-time implementation method for software-in-the-loop (SIL) simulation for control systems, primarily for control education. The SIL simulation is carried out by using a PC for the controllers, a PC for the plant, an open network, and a general-purpose computer-aided control system design (CACSD) package. Specially, Ethernet network is investigated in terms of control issues such as sampling interval, network-induced time-delay, use with many I/O points and data synchronization. A performance evaluation of software-in-the-loop simulation is made with respect to a computation delay and a sampling interval. To reduce the effects of the time-delay, particularly for fast plants, we introduce a time-scaling method that leads to a slow motion. It is demonstrated that this real-time SIL simulation will be very useful, particularly for control education.*

Keywords: Software-in-the-loop simulation, Control education, Computer-aided control system design (CACSD)

1. **Introduction.** The design and analysis of control systems require algebraic or numerical computations, various plots, and commonly used mathematical algorithms 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 [27-30]. 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 located apart from each other and exchange, in real-time, several analog signals for continuous variables and a few digital signals for event variables. However, most CACSD packages run in a single personal computer (PC), where a plant and a controller are simulated together and hence some practical problems from their connection or communication cannot be simulated. These existing packages are not real timed and far from reality.

An approach to simulate real control systems is to provide some systems in which the simulated plant and the simulated controller can be connected to each other and run in two personal computers in real-time. We call the method software-in-the-loop (SIL) simulation compared with the hardware-in-the-simulation [9-11], because only software exists in the control loop. A SIL simulation is particularly useful for control education because real control experiment systems are too expensive for large classes in big universities.

For real-time SIL simulation, conventional PCs will be suitable for its low cost and easy use, together with a general-purpose CACSD package for easy programming for control algorithms. Open networks are also necessary for replacing multiple connections of many 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 systems where the analog signal exists [13].

For real-time SIL simulation, the control-oriented requirements such as data synchronization, multiple I/O handling, and data transmission speed for a short sampling interval must be carefully considered with network-oriented issues such as scheduling, media access control and media transmission delay. There exist few discussions of these issues for real-time SIL simulation. Even though many CACSD packages have been developed [1,2,5,7,14-16], CACSD packages suitable for real-time use of two PCs with open networks do not exist to the authors' knowledge. This paper investigates how to achieve control-oriented requirements in commonly used Ethernet network. For easier programming and connection with network, a block-oriented network box and a specialized software are also suggested in this paper. This network box is implemented and added to an existing CACSD package, CEMTool/SIMTool, which was developed at Seoul National University, Korea. As the main advantage of real-time SIL simulation proposed in this paper, it is helpful for students to understand the independent behaviors of plants and controllers at low cost without expensive control experiment systems.

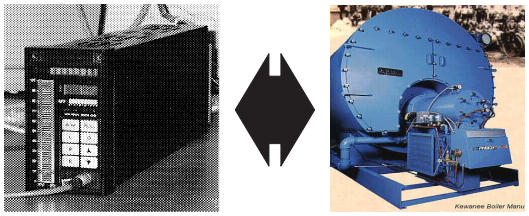
A SIL simulation inevitably has computation and network delays for both the controller and the plant and thus a SIL simulation for fast plant may not be useful. If fast plants cannot be dealt with effectively by this simulation, a slow motion function is suggested to handle these fast plants. The slow motion function is achieved by using a time-scaling method and its usefulness is shown via simulation.

For control education, many models need to be implemented in a model toolbox so that users can select a suitable model for experiments. The suggested real-time SIL simulation can be carried out in a computer laboratory with many PCs or even in a classroom with two notebook computers. How it can be used in a real class setting will be discussed.

This paper is organized as follows: in Section 2, real-time SIL simulation and its requirements are introduced; in Section 3, communication characteristics of Ethernet and the network with AD/DA among four networks are investigated; in Section 4, performance evaluation of real-time SIL simulation is carried out, subject to computation and network delays of the plant and the controller; in Section 5, 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 6, a network and model boxes are suggested for easy programming of networks; in Section 7, experiment results are shown for a slow plant and a fast plant using the network; in Section 8, how to apply the method proposed in this paper to control education is shown; finally, our conclusions follow in Section 9.

2. **Real-time Software-In-the-Loop (SIL) Simulation and Requirements.**

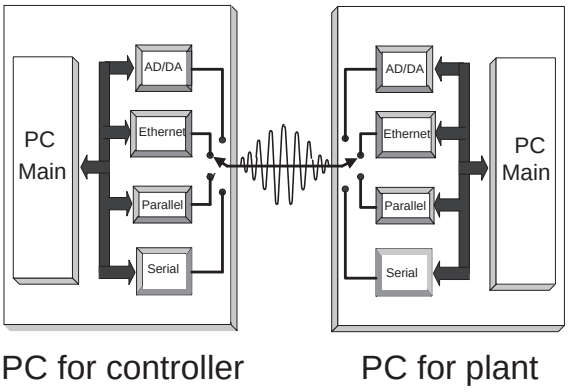
Consider a real control system, Figure 1(a), in which the left side shows a loop controller that is very widely used in industry as the core device of controllers, and the right side shows a boiler plant. The separated controller and plant can be modeled by two separated PCs (Figure 1(b)) with an open 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.



Analog or digital controller

Real plant

(a) Real control system and real plant



(b) SIL with networks

FIGURE 1. Real-time SIL simulation system using two PCs and a network

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 multi-variable 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 easily linked 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.

3. **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. Analysis of serial and parallel network will not be discussed for simplicity.

1) Ethernet network: 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 2 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 receives 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.

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.

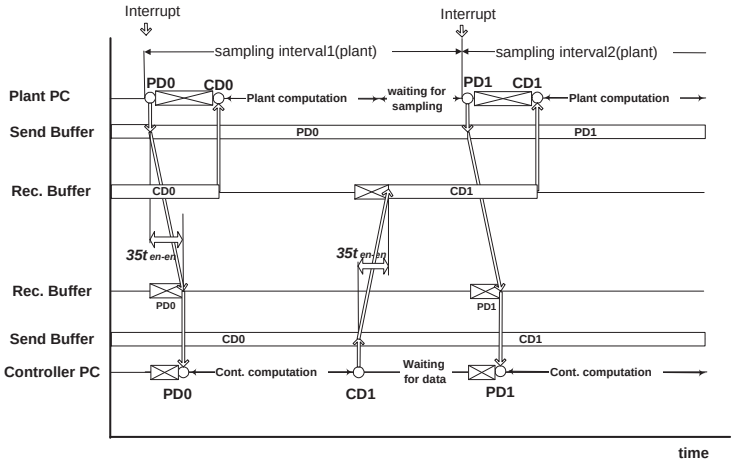


FIGURE 2. Ethernet network timing chart for one channel

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.

3) Comparison: A comparison of four networks is shown in Table 1. We can see that the Ethernet network may be better than the AD/DA network 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.

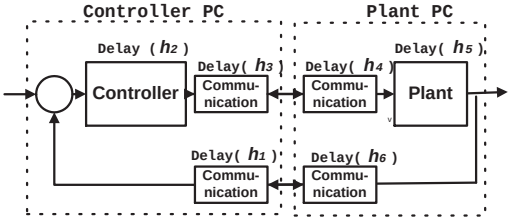
4. **Total Delay Time In-the-Loop and Performance Evaluation.** In a real-time SIL simulation, both the plant PC and the controller PC have computation and communication delays because of the computation of dynamics and the communication of the input and the output data. These delays can make the performance of the controlled system worse or even make it unstable [18,19]. Hence, it is necessary to evaluate the effect of delays in these systems.

Both plant and controller delay in a SIL simulation can be considered as (Figure 3(a)). The delay in a real-time SIL simulation, especially for the Ethernet network case, is modeled in Figure 3(b), which shows the controller delay and the plant delay. As we can see in Figure 3(b), the controller and plant have a date-input delay (h_1, h_4), computation delay (h_2, h_5) and a data-output delay (h_3, h_6). The Ethernet network always has one

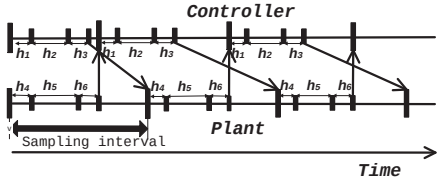
TABLE 1. Evaluation of two networks as education tools

Interface	Ethernet	AD/DA	Serial	Parallel
Fast plant experiment	good	good	good	good
Slow plant experiment	good	good	long	medium
Time delay	short	medium	slow	medium
Communication time for multiple I/O points	slow	medium	short	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

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.



(a) Plant and controller delays in SIL



(b) Total delay in Ethernet

FIGURE 3. Time delays in SIL

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

$$\dot{h_1} = -\frac{1}{R_1 C_1} h_1 + \frac{1}{R_1 C_1} h_2 + \frac{1}{C_1} W \tag{1}$$

$$\dot{h_2} = \frac{1}{R_1 C_2} h_1 - \left(\frac{1}{R_1 C_2} + \frac{1}{R_2 C_2} \right) h_2 \tag{2}$$

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 $h_1(0) = 100m$, $h_2(0) = 100m$, $R_1 = 0.009\text{sec}/m^2$, $C_1 = 10m^2$, $R_2 = 0.2\text{sec}/m^2$ and $C_2 = 3m^2$. The time constant of this system is 0.0402. 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 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 4. In this simulation, the sampling interval of both the plant and the controller is 0.055sec.

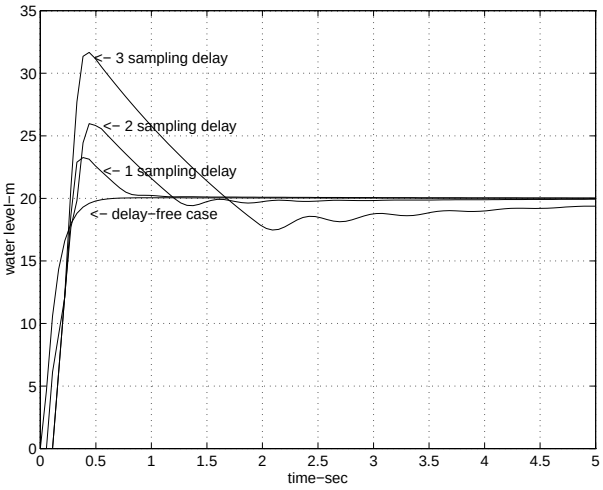


FIGURE 4. Effect of delay in a SIL simulation

5. **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 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. It is noted that if this kind of time scaling method is employed, we can not make a real time simulation, but get an accurate result.

Let an original system be defined as follows:

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

A new time variable ($t' = \alpha t$) is introduced for a positive real value α. Then, the original system becomes

$$\begin{aligned} \frac{dx(\frac{1}{\alpha}t')}{dt'} &= \frac{1}{\alpha} f\left(x\left(\frac{1}{\alpha}t'\right), u\left(\frac{1}{\alpha}t'\right)\right) \\ y\left(\frac{1}{\alpha}t'\right) &= g\left(x\left(\frac{1}{\alpha}t'\right)\right). \end{aligned}$$

For a controller such as:

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

the time-scaled controller equation becomes

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

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

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:

(a) Delay-free controller and delay-free plant.
(b) Real-time SIL simulation: plant and controller delays (Plant delay = 0.055sec, Controller delay = 0.055sec).
(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 4.

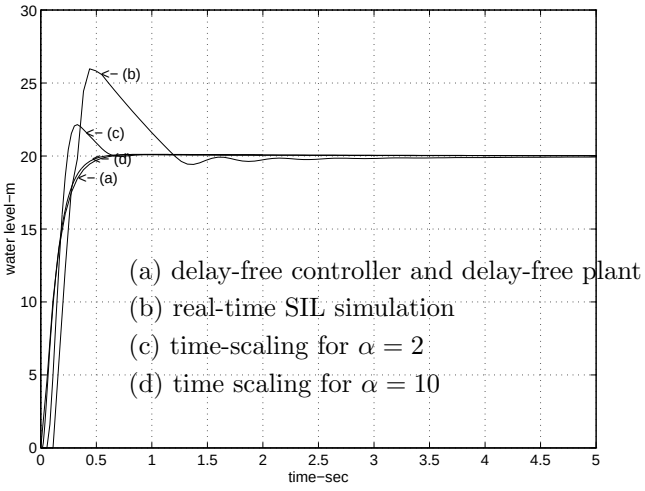
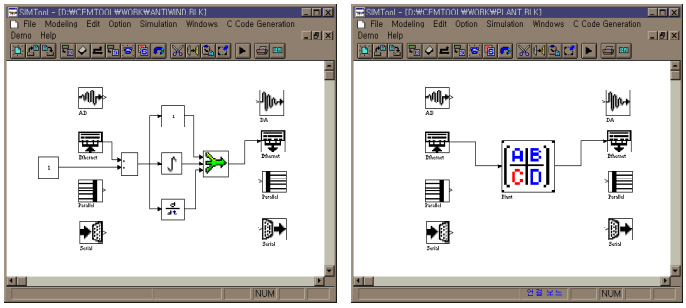


FIGURE 5. Comparison of performance

Figure 5 shows step responses for control systems (a) – (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 with (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.

6. **A Suggested Network and Model Boxes.** 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 [20,21] block diagram graphic editor [22,25], compiler and executor [26], with which a user can implement both plants and controllers. Figure 6 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 six parameters such as host name, peer name, channel number, high input, low input and time-scaling coefficient; for



(a) Controller (b) Plant

FIGURE 6. Controller and plant modeled with network blocks

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 7.

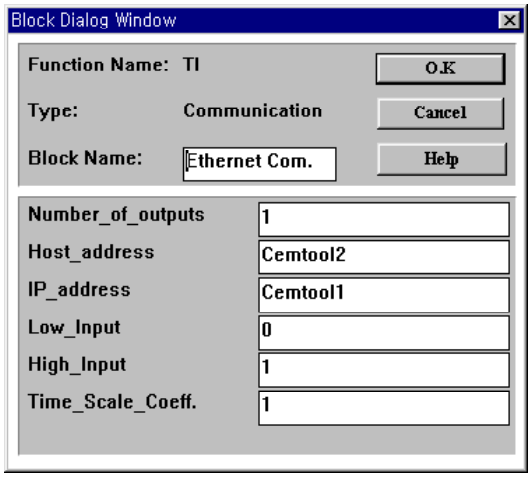


FIGURE 7. Input tuning parameters for an ethernet network

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 8. 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.

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 9.

7. **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 10. It is programmed with CEMTool/SIMTool in the controller PC, which has a gain block, an

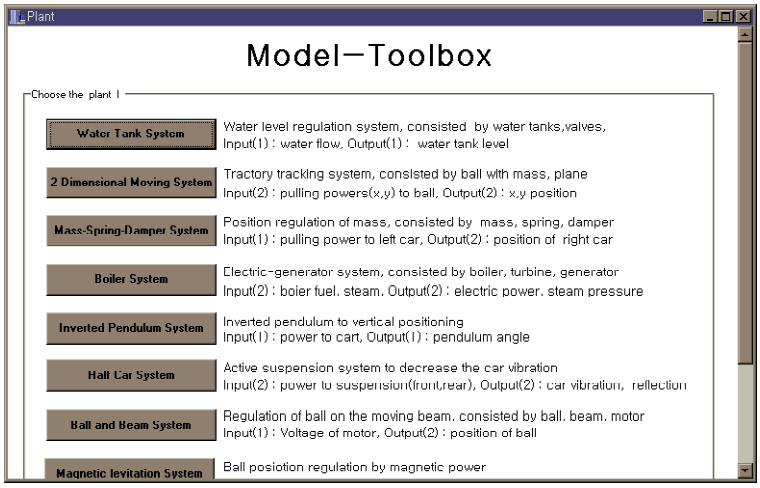


FIGURE 8. A part of the model toolbox window

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 $20m$. 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 0.055sec, which is set as one clock-tick time in the controller PC. The plant PC is synchronized with this sampling interval.

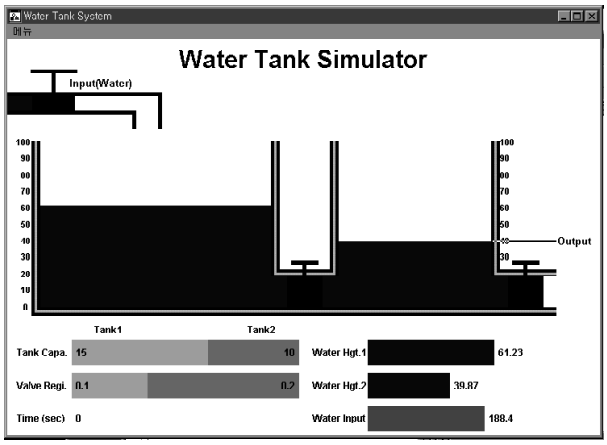


FIGURE 9. Animated water tank for a plant

The controller PC communicates with the plant PC by selecting one of the four different networks, as in Figure 6. For the following experiments, Ethernet communication between the controller PC and the plant PC is used.

Figure 11 shows the step response for a slow water tank system. In this experiment, the specifications of the water tanks are that $h_1(0) = 100m$, $h_2(0) = 100m$, $R_1 = 0.01\text{sec}/m^2$, $C_1 = 200m^2$, $R_2 = 0.01\text{sec}/m^2$ and $C_2 = 100m^2$. The time constant of this system is 0.8. The eigenvalues of the system matrix from the models (1) and (2) are computed as -0.2192 and -2.2808. Case (a) in Figure 11 shows the response of the delay-free system, which is computed in a single PC without a network delay. Case (b) in Figure 11 shows the response of the real-time SIL simulation. It can be seen that the real-time SIL simulation offers almost the same as the delay-free system.

Figure 12 shows a step response for a fast water tank system. In this SIL simulation, the specifications of the water tanks are $h_1(0) = 100m$, $h_2(0) = 100m$, $R_1 = 0.0009\text{sec}/m^2$,

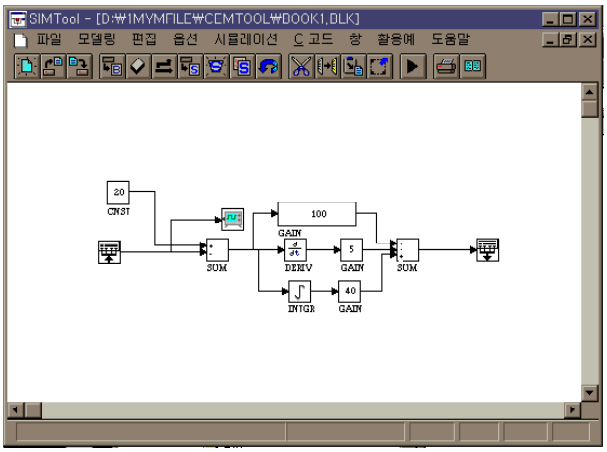


FIGURE 10. SIMTool blocks for a PID controller

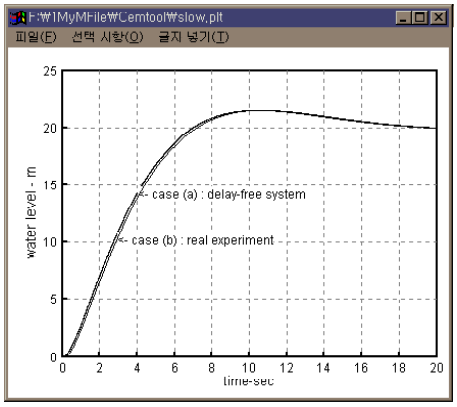


FIGURE 11. Real-time SIL simulation for a slow water tank system

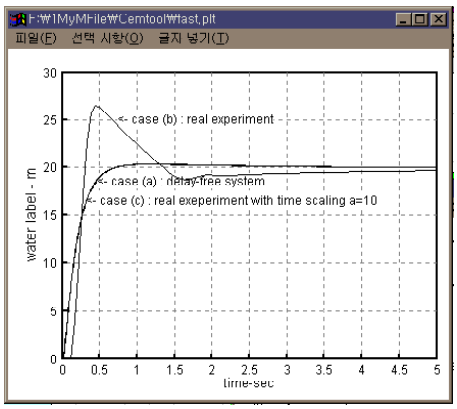


FIGURE 12. Real-time SIL simulation for a fast water tank system

$C_1 = 10m^2$, $R_2 = 0.2\text{sec}/m^2$ and $C_2 = 3m^2$. The time constant of this system is 0.0402, which is smaller than the sampling interval 0.055sec. The eigenvalues of the system matrix from the models (1) and (2) are computed as -0.3836 and -482.7646. Case (a) in Figure 12 shows the response of the delay-free system. Case (b) in Figure 12 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 12 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.

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

8. **Guidelines for Control Education.** The tendency of the control educational world is that the theory based on text books runs in parallel with experiments. Usually, the course on control consists of lecture and practical trainings spending about two hours a week each. Since the real control experiment devices for practical training are very expensive in general, particularly for large classes in big universities, a computer laboratory equipped with several personal computers can be used for experiment environment with low costs to many students.

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. In case of very fast plants, time-scaling method can be recommended to students. 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.

9. **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, the four networks 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. Most students were full of interests for experiments with the real-time SIL simulation and got the feel of the control design under the more realistic environments.

Acknowledgment. This paper was supported by Konkuk University in 2009. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

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, China, pp.59-64, 1988.

[2] A. Boom, CADCS developments in Europe, *IFAC Computer Aided Design in Control Systems*, Beijing, China, pp.65-74, 1988.

[3] M. Rimvall, Computer-aided control system design, *IEEE Control Systems Magazine*, pp.14-16, 1993.

[4] M. Andersson, D. Bruck, S. E. Mattsson and T. Schonthal, OmSim – An intergrated interactive environment for object-oriented modeling and simulation, *IEEE/IFAC Joint Symposium on CACSD*, Tucson, AZ, pp.285-290, 1994.
[5] B. A. Ogunnaike, The role of CACSD in contemporary industrial process control, *IEEE Control Systems Magazine*, vol.15, no.2, pp.41-47, 1995.
[6] R. Rutz and J. Richert, CAMeL: An open CACSD environment, *IEEE Control Systems Magazine*, vol.15, no.2, pp.26-33, 1995.
[7] J. James, F. Cellier, G. Pang, J. Gray and S. E. Mattsson, The state of computer-aided control system design (CACSD), *IEEE Control Systems Magazine*, vol.15, no.2, pp.6-7, 1995.
[8] M. G. Safonov and R. Y. Chiang, CACSD using the state-space L-infinity theory – A design example, *IEEE Transactions on Automatic Control*, vol.33, no.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, *The 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, *The 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] K. H. Grantham, Issues in the development of the interactive CACSD package SFPACK, *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, CA, pp.250-254, 1992.
[16] M. Wette, Casey: A computer-aided engineering system, *IEEE Symposium on CACSD*, Napa, CA, pp.232-237, 1992.
[17] W. H. Kwon, *Practical Automatic Control Using CEMTool*, Cheng Moon Gock, 1996.
[18] 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.
[19] R. Krtolica, U. Ozguner, H. Chan, H. Goktas, J. Winkelman and M. Liubakka, Stability of linear feedback system with random communication delays, *International Journal of Control*, vol.59, no.4, pp.925-953, 1994.
[20] C. P. Jobling, P. W. Grant, H. A. Barker and P. Townsend, Object-oriented programming in control-system design – A survey, *Automatica*, vol.30, no.8, pp.1221-1261, 1994.
[21] R. Kroeger, Using object-oriented programming standards for developing distributed real-time control applications, *The 3rd IFAC/IFIP Workshop on AARTC'95*, Ostend-Belgium, pp.135-153, 1995.
[22] M. Rimvall, Interactive environments for CACSD software, *IFAC Computer Aided Design in Control Systems*, Beijing, China, pp.17-26, 1988.
[23] 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*, vol.25, no.2, pp.311-316, 1989.
[24] D. Xue and D. P. Atherton, BLOCKM – A block diagram modeling interface and its applications, *IEEE Symposium on CACSD*, Napa, CA, pp.242-249, 1992.
[25] O. Ravn and M. Szymkat, Requirements for user interaction support in future CACE environments, *IEEE/IFAC Joint Symposium on CACSD*, Tucson, AZ, pp.381-386, 1994.
[26] M. Koga, M. Sampei and K. Furuta, A compiler of MATLAB to MATX compiling and linking of m-files to an executable program, *IEEE/IFAC Joint Symposium on CACSD*, Tucson, AZ, pp.381-386, 1994.
[27] K.-C. Yao and C.-H. Hsu, Robust optimal stabilizing observer-based control design of decentralized stochastic singularly-perturbed computer controlled systems with multiple time-varying delays, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.467-478, 2009.
[28] K.-C. Yao, C.-Y. Lu, W.-J. Shyr and D.-F. Chen, Robust output feedback control design of decentralized stochastic singularly-perturbed computer controlled systems with multiple time-varying delays, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4407-4414, 2009.

[29] K.-C. Yao, Robust output feedback stabilizing computer control of decentralized stochastic singularly-perturbed systems, *ICIC Express Letters*, vol.1, no.1, pp.1-7, 2007.

[30] K.-C. Yao and F.-Q. Cai, Robustness and reliability of decentralized stochastic singularly-pert urbed computer controlled systems with multiple time-varying delays, *ICIC Express Letters*, vol.3, no.3(A), pp.379-384, 2009.