—Brief Paper—

INTERNET-BASED REAL-TIME CONTROL ARCHITECTURES WITH TIME-DELAY/PACKET-LOSS COMPENSATION

Kun Ji, Won-jong Kim, and Abhinav Srivastava

ABSTRACT

The main objective of this paper is to demonstrate the feasibility of Internet-based real-time control. A novel client/server-based architecture for Internet-based supervisory control with a Common Gateway Interface/Hyper Text Markup Language (CGI/HTML) interface is presented. A real-time operating environment was established for closed-loop control over Ethernet. We conceived of an autoregressive (AR) prediction scheme and a novel compensation algorithm to compensate for network-induced time delays and data-packet losses simultaneously. We constructed an open-loop unstable ball magnetic- levitation (maglev) setup as a test bed to validate the two proposed control architectures. Experimental results proved the feasibility of Internet-based real-time control and verified the effectiveness of the proposed time-delay/packet-loss compensation algorithm in networked feedback control systems.

KeyWords: Internet-based supervisory control, networked feedback control, time-delay/packet-loss compensation.

I. INTRODUCTION

Distributed real-time control can be roughly classified into two modes: (1) tele-operation and supervisory control, and (2) networked feedback control. Internet-based tele-operation and supervisory control were used in tele-robotics, remote manufacturing, tele-surgery, and distant education [1,2]. Especially, the Mercury project developed by Goldberg *et al*. was the first successful use of the Internet for supervisory control of an Internet-based robot [1]. For networked feedback control, researchers proposed control structures to mitigate the detrimental effect of data-transmission delays and communication failure. Ray and Halevi [3,4] proposed an augmented deterministic discrete-time methodology to control a linear plant over a periodic delay network. Walsh *et al*. [5] used a nonlinear and perturbation theory to formulate network delay effects in a networked control system (NCS). Krotolica *et al*. [6] designed a networked controller in the frequency domain using robust control theory. Yook *et al*. [7] proposed a framework for NCSs in which estimators are used at each node to save the bandwidth.

In this paper we present a novel hardware and software architecture for Internet-based supervisory control in Section II. Section III presents an architecture for networked feedback control along with experimental verification. Conclusions are given in Section IV.

II. INTERNET-BASED SUPERVISORY CONTROL

Paschall [8] developed a ball maglev system. The objective of this maglev system is to levitate a steel ball at a predetermined steady-state equilibrium position with an electromagnet. The framework of our supervisory control via the Internet is shown in Fig. 1. As a test bed of our Internet-based supervisory control, the ball maglev setup is connected to the host Pentium IV personal computer (PC) that runs the Internet Information Services (IIS) 5.0 Web server on the Windows 2000 Professional operating system (OS).

Manuscript received February 4, 2005; revised June 27, 2005; accepted November 27, 2005.

K. Ji, W.-J. Kim are with the Department of Mechanical Engineering Texas A&M University, College Station, TX 77843-3123, U.S.A. (e-mail: {kunji, wjkim}@tamu.edu).

A. Srivastava is with the Arbin Instruments, College Station, TX 77845, U.S.A.

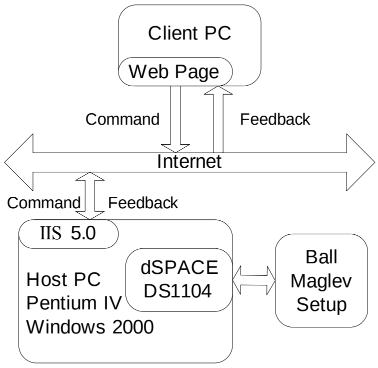

Fig. 1. Framework of the supervisory control via the Internet.

The host PC runs a Web server that can serve Web pages related to the test bed. Once the commands have been submitted to the Web page, the Web page transfers these commands via the Internet to the Web server on the host PC. The transmission of these commands usually takes place by the Transmission Control Protocol/Internet Protocol (TCP/IP). The Web server on the host PC passes these commands to the CGI script present in the CGI bin of the Web server. The CGI script is executed in the CGI bin as soon as the request from the client is received. This CGI script is used to convert the encrypted data from the client into a format understandable by the host PC. These commands from the CGI script are then transferred to the controller board in real time using C programs. The results of the performance of the test bed with user-defined commands are then sent via the Internet in the opposite direction using the same methodology.

The real-time digital control algorithm was written in the C programming language and implemented on a dSPACE DS1104 DSP board so that it can easily communicate with the CGI environment. This real-time C code is interrupt driven and is called in the very beginning of each sampling period [9].

With the integration of the Internet in this supervisory control scheme, the maglev system can now be accessed from anywhere via the Internet. The client PC can give the position commands remotely through the Internet Web page to move the steel ball within its travel range and get real-time control results. A 300-μm step response obtained this way is shown in Fig. 2. The novelty of this supervisory-control architecture is that we use common web-browser software, not dedicated interface software. Thus this architecture has more practical engineering merit compared with other internet-based supervisory-control architectures.

In the supervisory control of the maglev system, the system stability was not much affected by the Internet. This was due to the fact that no sensor or control data traveled through a communication network and real-time control was achieved locally with the dSPACE controller board.

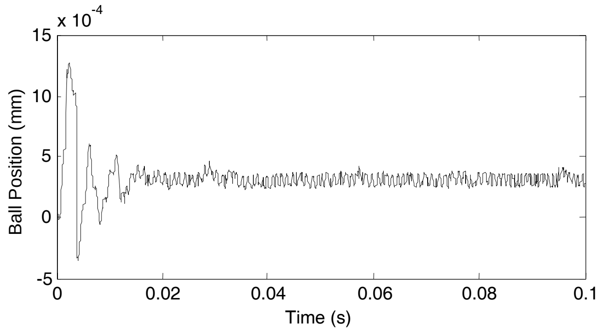

Fig. 2. Maglev system response to a step input of 300μm provided via the Internet.

III. NETWORKED FEEDBACK CONTROL

In this section, we consider networked feedback control. A block diagram of a networked feedback control system is shown in Fig. 3 where the control loop is closed through the network, which introduces time-varying delays τ_{sc} between the sensor node and the controller node and τ_{ca} between the controller node and the actuator node.

3.1 Experimental Determination of Time Delays

As in a typical communication network, sporadic surges in time delays were observed in the Local Area Networks (LANs) in our labs and are shown in a delay profile in Fig. 4. They were generated for various reasons, such as sporadic congestion in the network, use of bandwidth-intensive applications, and other users using the network capacity.

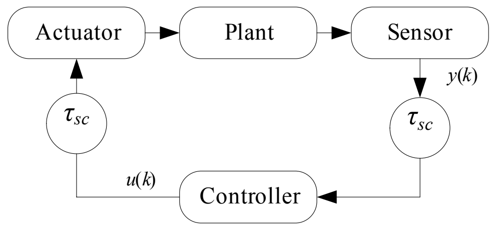

Fig. 3. Block diagram of a networked feedback control.

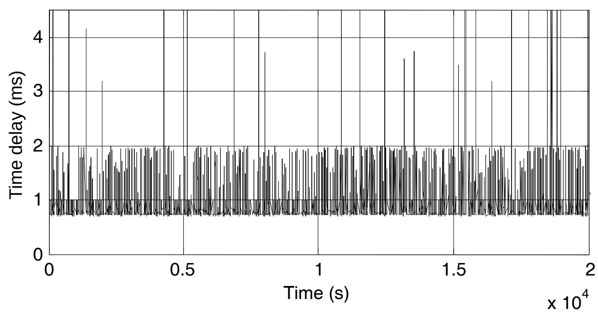

Fig. 4. Round-trip time delay profile between two PCs connected to two separate LANs in our labs.

3.2 Real-Time Operation Environment

The ball maglev system shown in Fig. 1 is used again as the test bed for network feedback control in this section. The ball maglev system is open-loop unstable, and the events of the sensor sampling the data and the actuator actuating the control have certain deadlines. If these deadlines are missed because of the indeterministic OS activities and network-induced time delays or data-packet losses, the system stability will be lost [10]. Thus a real-time operating environment is needed to ensure these time-constrained events do not miss their deadlines. Ambike developed a real-time control system for the ball maglev system using real-time application interface (RTAI) with Linux [10]. National Instruments' PCI-6025E is the data-acquisition card for the experiments. Figure 5 shows the framework of our networked feedback control system test bed with the ball maglev setup.

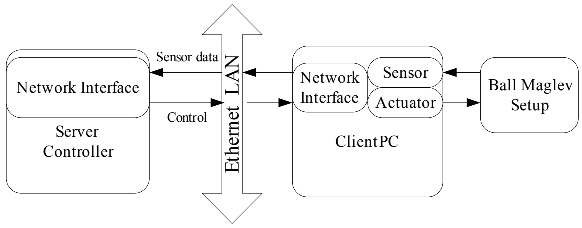

Fig. 5. Framework of the networked feedback control system.

3.3 Time-Delay/Packet-Loss Compensation

The stability of the ball maglev system can be lost because of the presence of sporadic surges in communication time delays in the LANs in our labs shown in Fig. 4. Based on our previous work [11], we developed a compensation algorithm to deal with these network-induced time delays and data-packet losses in both the feedback and feedforward paths simultaneously.

3.3.1 AR Model for Multi-Step-Ahead Sensor-Data and Control-Data Prediction

The plant and controller dynamics are modeled as

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) \tag{1}$$
$$u(k) = r(k) - Ky(k)$$

where $x \in R^n$ is the state, $y \in R^m$ is the output, $u \in R^v$ is the control input, and $r \in R^v$ is the command input. A, B, and C are constant matrices of compatible dimensions, and the output-feedback controller is represented with a gain matrix $K \in R^{v \times m}$.

To compensate for the delay τ_{sc}, we developed a prediction algorithm. Many experiments were conducted to select the best model to be used for sensor data prediction. The performances of these predictors were compared in [12]. An AR model was finally chosen because the percentage error variation for this model is less than others. The AR model is defined as

$$A(q^{-1})y(k) = u(k) \tag{2}$$

where q^{-1} is the backward shift (or delay), $y(k)$ and $u(k)$ correspond to the k-th output and input, and $A(q^{-1})$ is defined as

$$A\,(q^{-1}) = 1 - a_1 q^{-1} - \ldots - a_n q^{-n} \tag{3}$$

Based on the recursive least-square methodology, an off-line identification of the parameters of the fifth-order AR model was performed using MATLAB. After conducting many experiments with different orders, we observed that the models with orders greater than five had the percentage errors (as shown in Fig. 7) comparable to that achieved by the fifth-order AR model. Since higher-order models need more computation time, a fifth-order AR model was chosen for the sensor-data prediction and was sufficiently accurate with minimal execution time in the control loop. Figure 6 shows the convergence of the parameters in the off-line system identification process with real delay data. The parameters were determined as the mean values of the ranges in which these parameters converged.

Another independent set of real sensor data was collected to validate this model. The percentage error between the predicted values and the actual values of the sensor data was less than 4% as shown in Fig. 7.

Predictors were also designed for up to 4-step-ahead prediction of the sensor data. The 4-step-ahead prediction for the output-feedback control-data packets (u_1, u_{2p}, u_{3p}, u_{4p}, and u_{5p}) was then performed using the predicted sensor-data packets (y_1, y_{2p}, y_{3p}, y_{4p}, and y_{5p}) as

$$u_{kp} = r - Ky_{kp}, \qquad k = 2, \ldots, 5\ . \tag{4}$$

This method is also effective for the time-varying delay as long as it is bounded. For instance, if the delay is bounded by p sampling periods, then a p-step-ahead prediction method can be used for the time-delay compensation.

3.3.2 Time-Delay τ_{ca} and Packet-Loss Compensation

To compensate for the time-delay τ_{ca}, a compensation algorithm was developed. Figures 8 and 9 show two examples of the time-delay and data-packet-loss compensation. In these figures, the label y denotes the sensor data transferred from the client sensor to the server controller, and the label u, the control signal data transferred from the server controller to the client actuator. The subscripts of these labels denote the sampling-period indices and indicate whether the data are predicted (p). For example, y_2 is the sensor data of the second sampling period, u_3 is the control data for the third sampling period, and u_{4p} is the predicted control data for the fourth sampling period.

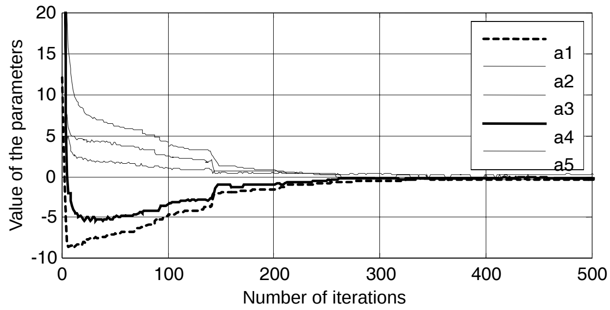

Fig. 6. Convergence of the parameters with respect to the number of iterations. All the parameters converged after 350 iterations.

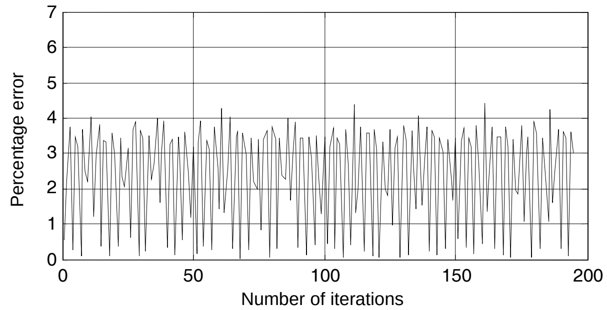

Fig. 7. Percentage error between the predicted and actual sensor data.

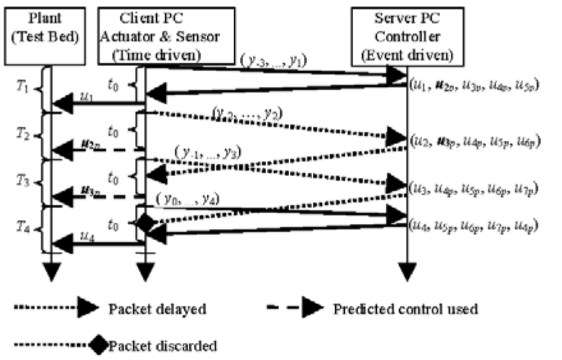

Fig. 8. Communication process with two-way time delays.

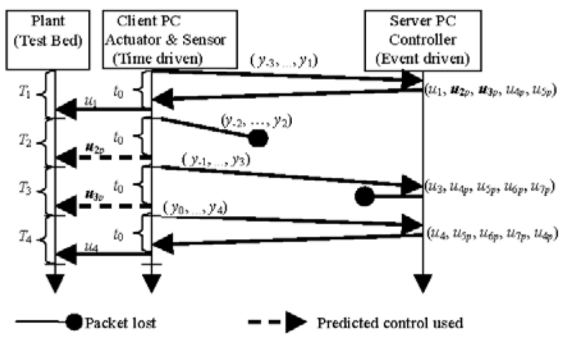

Fig. 9. Communication process with two-way packet losses.

In our network communication, all data-packets are time stamped. The round-tip arrows represent lost data communication in a given sampling period, and the dotted arrows, delayed communication. The dashed arrows indicate that the formerly predicted control input is applied when the actual current control-data packet does not reach the client side in time. The symbol t_0 represents the time threshold. The square-tip arrows indicate that the delayed control-data packet of the previous sampling period is discarded if there is a new control-data packet available. Thus most recent control data such as u_4 shown in Fig. 8 are used. Thus the proposed compensation algorithm can deal with time delays and packet losses in both the feedback and feedforward paths simultaneously. In case of out-of-order transmission of packets, the outdated packets are simply discarded. Thus a control input to the plant is always generated in each sampling period with either actual or predicted control data depending on the actual data's availability.

3.3 Experimental Verification

Our ball maglev setup is open-loop unstable, thus it is suitable for the verification of our compensation algorithm developed above. The effect of p consecutive packet losses is equivalent to that of time delays as long as p sampling periods in our compensation algorithm. Three experiments were conducted. In the first experiment, no compensation was used. At t = 12 s, we forced the sensor-data packet to be lost while transferred from the client to the server, then a zero control input was applied to the actuator. The response of the system is shown in Fig. 10. The 0 value of the y-axis indicates that the ball could not maintain its equilibrium position and fell down.

In the second experiment, the compensation algorithm was used and 4 successive sensor-data-packet losses occurred every 12 s from t = 2 s onwards. The response of the system is shown in Fig. 11. From Fig. 11, we can see that the system remained stable throughout the experiment, and the ball did not fall down from its equilibrium position. This experimental result demonstrates that our algorithm is effective to maintain the system stability with up to 4 successive packet losses or time delays as long as 4 sampling periods.

In the third experiment, the compensation algorithm was implemented. From t = 12 s onwards, artificial packet loss was introduced at every fifth sample (*i.e.* 20% packet-loss rate). The response of the system is shown in Fig. 12. The system remained stable throughout the experiment, and the ball did not fall down from its equilibrium position. However, the fluctuation in the ball motion about the equilibrium point increased. This performance degradation resulted from the 20% packet loss.

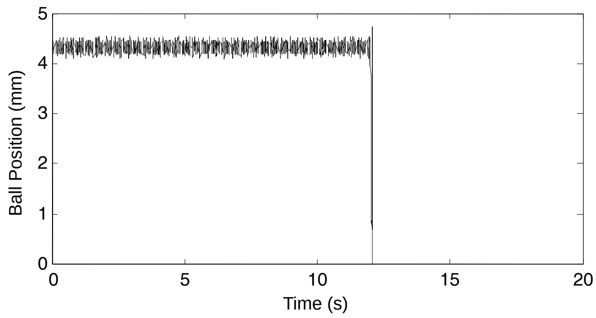

Fig. 10. Ball position profile with packets loss occurring at t = 12 s without compensation.

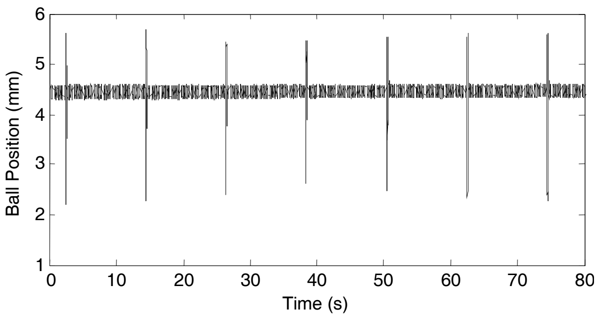

Fig. 11. Ball position profile with 4 successive packet losses occurring every 12 s after t = 2 s.

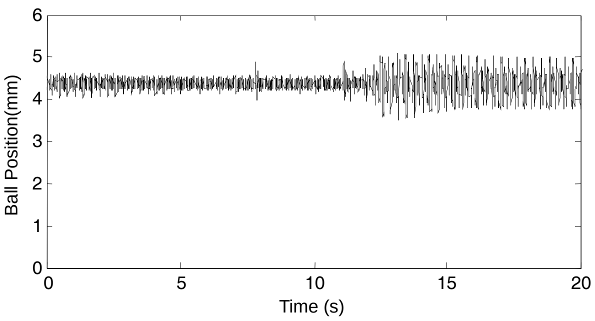

Fig. 12. Ball position profile with average 20% packet loss occurring after t = 12 s.

IV. CONCLUSIONS

In this paper, we validated the Internet-based supervisory control and networked feedback control architectures experimentally using an open-loop unstable ball maglev system. First we developed a client/server architecture for the Internet-based supervisory control. Second, a novel Ethernet-based delay- and packet-loss-compensation methodology for networked feedback control was presented in this paper. Its main objective was to compensate for the time delays and data-packet losses in the network communication simultaneously. With the ball maglev system as the test bed, we experimentally verified that the methodology proposed herein ensured the closed-loop system stability even in the presence of bounded sporadic surges in time delays up to four sampling periods or four successive data-packet losses.

REFERENCES

1. Goldberg, K., S. Gentner, and J. Wiegley, *The Mercury Project: A Feasibility Study for Internet Robots*, Technical Report, UC Berkeley and University of Southern California (1994).
2. Garcia, C.E., R. Carelli, J.F. Postigo, and C. Soria, “Supervisory Control for a Telerobotic System: A Hybrid Control Approach,” *Control Engineering Practice*, Vol. 11, No. 7, pp. 805–817 (2003).
3. Ray, A. and Y. Halevi, “Integrated Communication and Control Systems: Part I — Analysis,” *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 110, No. 4, pp. 367–373 (1988).
4. Ray, A. and Y. Halevi, “Integrated Communication and Control Systems: Part II — Design Consideration,” *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 110, No. 4, pp. 374–381 (1988).
5. Walsh, G.C., Y. Hong, and L.G. Bushnell, “Stability Analysis of Networked Control System,” *IEEE Transactions on Control System Technology*, Vol. 10, No. 3, pp. 438–46 (2002).
6. Krotolica, R., Ü. Özgüner, H. Chan, H. Göktaş, J. Winkelman, and M. Liubakka, “Stability of Linear Feedback Systems with Random Communication Delays,” *International Journal of Control*, Vol. 59, No. 4, pp. 925–953 (1994).
7. Yook, J.K., D.M. Tilbury, and N.R. Soparkar, “Trading Computation for Bandwidth: Reducing Communication in Distributed Control Systems Using State Estimators,” *IEEE Trans. Control Systems Technology*, Vol. 10, No. 4, pp. 503–518 (2002).
8. Paschall, II, S.C., *Design, Fabrication and Control of a Single Actuator Magnetic Levitation System*, Senior Honors Thesis, Texas A&M University (2002).
9. Boutell, T., *CGI Programming in C and Perl*, Addison-Wesley Developers Press (1995).
10. Ambike, A., “Closed-Loop Real-Time Control on Distributed Networks,” M.S. Thesis, Texas A&M University, College Station, TX (2004).
11. Ji, K., W.-J. Kim, and A. Ambike, “Control Strategies for Distributed Real-Time Control with Time Delays and Packets Losses,” *ASME International Mechanical Eng. Congress and Exposition*, Paper No. 61733 (2004).
12. Srivastava, A., “Distributed Real-Time Control via the Internet,” M.S. Thesis, Texas A&M University, College Station, TX (2003).