Letters

Network-Based Control With Real-Time Prediction of Delayed/Lost Sensor Data

Won-jong Kim, Kun Ji, and Abhinav Srivastava

***Abstract*—We present and experimentally verify two control architectures: supervisory control and feedback control over network. With the client/server architecture we developed for supervisory control, a client can give the control commands, tune the control parameters, and receive the experimental results remotely and in real time. Real-time feedback control over network is also investigated with a ball magnetic-levitation (Maglev) system as a test bed. We devised a novel timeout scheme and derived an autoregressive (AR) prediction model for delayed/lost sensor data. By applying this timeout and prediction scheme to the cases of consecutive data loss, we showed that the test bed could survive sporadic delays of as long as three sampling periods without losing stability. The feasibility and effectiveness of the proposed network-based real-time control methodologies was demonstrated experimentally.**

***Index Terms*—Communication delay, feedback control, packet loss, sensor data prediction, supervisory control.**

I. INTRODUCTION

THE Internet as a communication medium provides cost-effective, flexible, and easy-to-access means for distributed control systems. In supervisory control via the Internet the sensors, controllers, and actuators are located at the plant site, and the control loop is closed locally. Recently, researchers started using the Internet to establish supervisory control for their telerobots and test beds [1]–[3]. For real-time control via network, many methodologies have been formulated based on several types of network behaviors and configurations in conjunction with various ways to treat network delay [4]–[10]. A queuing methodology developed by Luck and Ray [4] used an observer to estimate the plant states and a predictor to compute the predictive control based on past output measurements. Nilsson modeled the random delays using Markov chains with the subsequent delays dependent on the preceding delay [8]. Ploplys and Alleyne developed a distributed-control system for a Furuta pendulum over a wireless communication network [10]. Although there is great potential in Internet-based distributed control, several technical challenges in performing real-time closed-loop control via the Internet should be addressed: (1) communication networks have inevitable time delays and data loss that are detrimental to real-time control; (2) the bursty behavior of communication networks causes sporadic surges in time delays; and (3) many users may share the finite bandwidth of the communication channel, and it is difficult to predict minute-by-minute data traffic.

To address the technical issues of network-induced time delays and data packet losses, we developed and experimentally verified two control architectures: supervisory control and feedback control over network. For supervisory control, we developed a new client/server architecture for supervisory control using a common gateway interface/hypertext markup language (CGI/HTML) interface, and made the control parameters be tunable via the Internet. For feedback control over network, we devised an algorithm for real-time prediction of delayed/lost sensor data in a general network data-traffic setting, and demonstrated that the control scheme developed in this paper could accommodate time delays 100 times longer than those without real-time prediction.

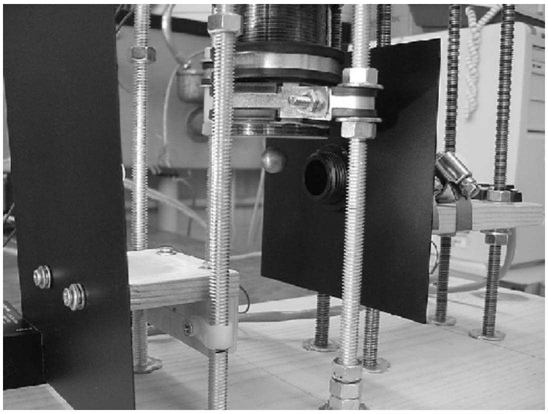

Fig. 1. Ball Maglev system [11].

II. NOVEL SUPERVISORY CONTROL VIA THE INTERNET

As reported in [11], Paschall and Kim developed a ball Maglev system shown in Fig. 1 to levitate a steel ball at a predetermined steady-state equilibrium position with an electromagnet. As a test bed of our Internet-based supervisory control, the ball Maglev setup is connected to the host personal computer (PC) as shown in Fig. 2. The Maglev system is controlled using a CGI/HTML interface with which a client can give the position commands remotely to move the steel ball within its travel range. The control parameters, such as the gain and the locations of poles and zeros can also be tuned on-line in real time. The client immediately receives the results from the changes made in the control parameters or commands.

A real-time control algorithm was implemented in software on a dSPACE DS1104 DSP controller board so that it can easily communicate with the CGI environment and obtain the corresponding system responses in real time from the Maglev test bed. The client can access this Web page of the Maglev system by typing the domain name of the host PC in the Web browser. An HTML page served by the host PC gets downloaded on the client PC so that the client can input his/her control parameters. Both the transactions of control parameters from the Internet and to the control algorithm are done simultaneously to save computation time. The software architecture for the supervisory control of the Maglev system via the Internet is depicted in Fig. 3.

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 need not be present at the lab to perform experiments. This network-based supervisory control methodology was successfully demonstrated in the Interactive Session at the 2003 American Control Conference while our Maglev test bed was being operated in College Station, TX [12]. This architecture has practical engineering merit with off-the-shelf software tools.

Manuscript received May 12, 2004; revised January 26, 2005. Manuscript received in final form July 27, 2005. Recommended by Associate Editor C. C. Lee. This work was supported in part by the National Science Foundation by Grant CMS-0116642.

W. Kim and K. Ji are with Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: wjkim@tamu.edu; kunji@tamu.edu).

A. Srivastava is with Arbin Instruments, College Station, TX 77845 USA (e-mail: abhinav@arbin.com).

Digital Object Identifier 10.1109/TCST.2005.859630

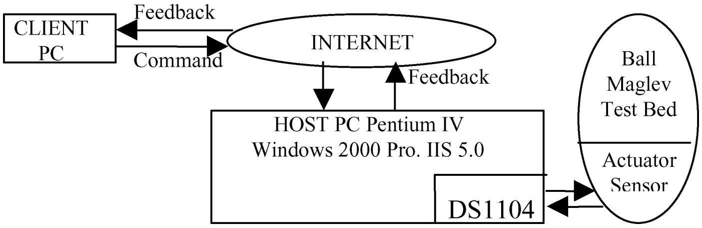

Fig. 2. Client/server architecture for supervisory control via the Internet.

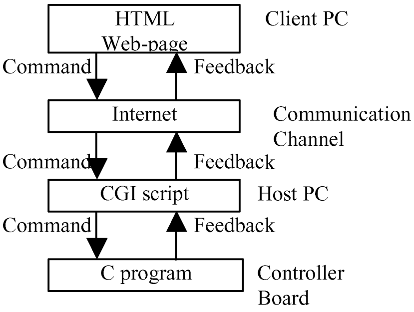

Fig. 3. Software architecture for the supervisory control of the Maglev system via the Internet.

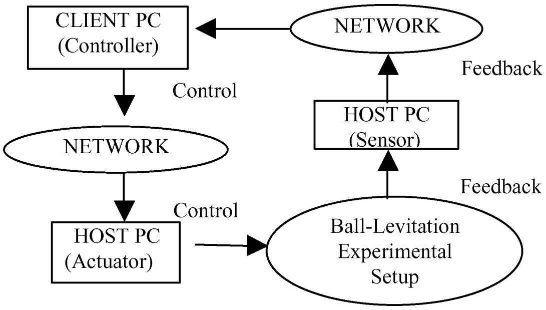

Fig. 4. Architecture for feedback control over network.

III. TIMEOUT SCHEME AND DELAYED/LOST SENSOR DATA PREDICTION FOR FEEDBACK CONTROL OVER NETWORK

After the successful establishment of the novel client/server architecture described in the previous section, we constructed a feedback control system shown in Fig. 4 in which the control loop is closed over the network. The host PC gets the sensor data and sends it to the client PC via the Ethernet. The client PC acts as the controller of the whole system. In this section, we analyze representative actual time delays in a feedback loop and propose a methodology to ensure system stability in the presence of bounded sporadic surges in communication time delays.

A. Experimental Determination of Time Delays

With a host PC and a client PC connected to separate local area networks (LANs) on the Texas A&M University campus, experimental delay data were collected using the Internet control message protocol (ICMP) [13], [14]. As in a typical network, sporadic surges in time delays were observed in our LANs as shown in a delay profile in Fig. 5.

B. Timeout Scheme

In our timeout scheme the controller computes the control input as soon as the new sensor data are available at the controller node. If no sensor data is available within the τ_0 timeout threshold, the controller calls a timeout and predicts the delaying/missing sensor data using a finite set of previous sensor data. A timing chart for our timeout scheme is presented in Fig. 6. In this figure, the two horizontal long arrows represent the time flow in the host PC and the client PC, respectively; two PCs communicate with each other via a LAN. The controller on the client side is both event-driven and time-driven, i.e., it calculates the control input as soon as the sensor data are available or the timeout is called. The short vertical solid line segments on the host-PC timeline represent sampling instants (with a period of T_s = 3 ms in our experiment). The short vertical dashed line segments on the client side mark the timeout threshold (1.42 ms used). The round-tip arrows indicate the lost sensor data communication in a given sampling period. The dashed arrows indicate the transmission of the predicted control input when the actual sensor data did not reach the client side before the timeout was called. Thus a control input to the plant is always generated in each sampling period with either actual or predicted sensor data depending on the actual data's availability.

C. AR Model for Delayed/Lost Sensor Data Prediction

The plant and controller dynamics are modeled as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \\ u(k) &= r(k) - Ky(k - h(\tau_k)) \end{aligned} \tag{1}$$

where the output feedback controller is represented with a gain matrix K. The finite nonnegative integer $h(\tau_k)$ represents the number of delayed/lost sensor data in the feedback, and can be represented by the following expression.

$$h(\tau_k) = \left\lfloor \frac{T_s + \tau_k - \tau_0}{T_s} \right\rfloor \tag{2}$$

where τ_0 represents the timeout threshold, τ_k, the time delay at the kth sample, and $\lfloor \cdot \rfloor$, the floor function. The controller waits for the latest sensor data for τ_0 after each sampling period commences before giving up for the availability of the actual sensor data.

For a given $h(\tau_k)$, the state vector in (1) can be written as

$$x(k+1) = A^{h(\tau_k)+1} x(k - h(\tau_k)) + \sum_{j=0}^{h(\tau_k)} A^j Bu(k-j). \tag{3}$$

Thus, from (1) and (3), we obtain

$$\begin{aligned} x(k+1) &= A^{h(\tau_k)+1} x(k - h(\tau_k)) \\ &\quad + \sum_{j=0}^{h(\tau_k)} A^j B[r(k-j) - k\hat{y}(k - h(\tau_k) - j)] \end{aligned} \tag{4}$$

where $\hat{y}$ represents an estimate of the sensor data. This estimated value is used whenever the communication delay exceeds the timeout threshold.

Among various methods, an AR model was used for the estimation of the sensor data $\hat{y}$ because of its simplicity [16]. Based on the recursive least-square methodology, an off-line identification of the parameters

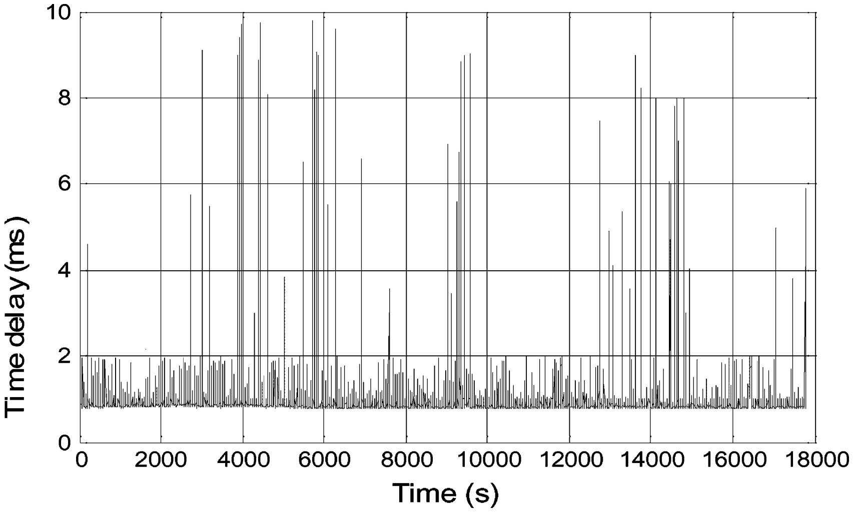

Fig. 5. Profile for round-trip time delays between the two PCs connected to two separate LANs.

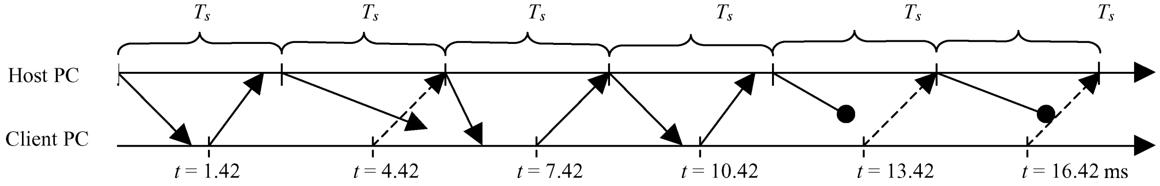

Fig. 6. Schematic timing chart that illustrates the timeout scheme.

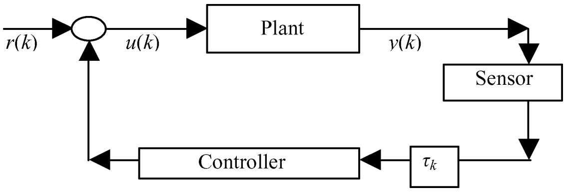

Fig. 7. Digital control system with communication delay τ_k.

of the fifth-order AR model was performed using MATLAB. The resulting fifth-order AR model used for the prediction of the sensor data was obtained as

$$\begin{aligned}\hat{y}(k) &= 0.3195y(k-1) + 0.0669y(k-2)\\ &\quad -0.0622y(k-3) + 0.1960y(k-4) + 0.4064y(k-5). \quad (5)\end{aligned}$$

It is assumed in (4) that the controller has access to all $h(\tau_k)$ previous sensor data. Since the nonnegative integer $h(\tau_k)$ varies from one timeout case to another, we extend this methodology to the cases when consecutive timeouts occur. If, for instance, the actual sensor data for the sampling instants $k, k-1, k-2$, and $k-3$ are not available (i.e., a case of four consecutive timeouts), the estimated values of sensor data for sampling instants $k-1, k-2$, and $k-3$ and the actual values for sampling instants $k-4$ and $k-5$ are used to predict $y(k)$ using (5).

Another independent set of real sensor data was collected and used to validate this model. The percentage error between the predicted values of sensor data using (5) and the actual values of the sensor data was less than 4%. Thus, we concluded that the predicted sensor data could be used for the calculation of the control input in the event of missing sensor data due to excessive communication delays or data packet losses.

Fig. 7 depicts a real-time digital control system using the Internet as a communication medium. The lumped delay τ_k is assumed to be present between the sensor and the controller [4], [15].

The server-based compensator for distributed delays in [4] requires knowing the accurate system model, which might not be practical for every case. In our approach, we use an AR model to predict the delayed/lost sensor data without requiring the knowledge of the system model and its states. Thus it is easier to implement. The following section verifies this approach experimentally.

IV. EXPERIMENTAL VERIFICATION FOR FEEDBACK CONTROL OVER NETWORK

A. Ball Maglev Test Bed and New Lead-Lag Controller

The ball Maglev system is used again as the test bed for feedback control over network. An original lead-lag compensator was designed and implemented using the matched pole-zero emulation method to stabilize the Maglev system [11] with the sampling frequency of 2 kHz.

$$G_C(z) = 3.33 \times 10^4 \left(\frac{z-0.966}{z-0.999}\right)\left(\frac{z-0.997}{z-0.705}\right). \quad (6)$$

From an actual measurement the upper bound of time delays for the existing lead-lag controller (6) was found to be 90 μs, and it is less than the mean time delay between the two PCs connected to the two separate LANs. Thus, a new lead-lag controller was designed directly in the digital domain using a zero-order-holder (ZOH) equivalence method to accommodate more time delay. The new implemented digital lead-lag compensator is

$$G_C(z) = 4.15 \times 10^4 \left(\frac{z-0.862}{z-0.923}\right)\left(\frac{z-0.892}{z+0.141}\right) \quad (7)$$

with a sampling frequency at as low as 333.3 Hz. When the time delay follows a uniform distribution, the new digital controller can accommodate a mean time delay of approximately 760 μs.

B. Application of the Timeout and Delayed/Lost Sensor Data Prediction Methodology

In our test bed, when a timeout occurs, the system goes unstable without performing any corrective action. The timeout and sensor data

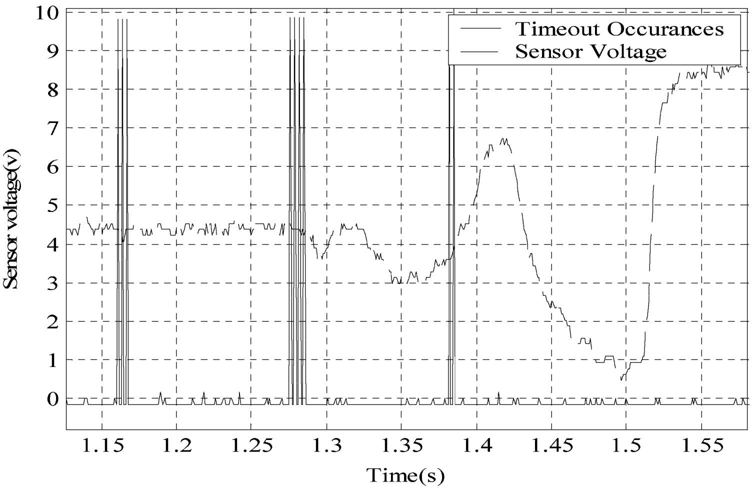

Fig. 8. Plots of sensor measurement data and corresponding timeouts with respect to time. The sensor voltage corresponds to the ball displacement.

prediction algorithm using (5) was implemented in a real-time control code to ensure the stability of the Maglev system for the delays following various probabilistic distributions. With the timeout threshold of 1.42 ms the mean time delay following a uniform distribution that can be accommodated in the control loop increased by 10% from 760 μs to 836 μs. To verify this methodology in the presence of sporadic surges in time delay, the round-trip time delay data from Fig. 5 were introduced in the control loop. As shown in Fig. 8, the sensor voltage output at the stable equilibrium was approximately 4.5 V that corresponds to the equilibrium position, and the digital controller (7) ran at the sampling frequency of 333.3 Hz. The vertical spikes of approximately 10 V in the figure indicate the timeouts called in the control code; a 10-V signal was outputted through another D/A channel when a timeout was called and recorded in a separate oscilloscope channel. The small ripples around the 0-V level are the noise in the D/A channel.

From the figure it can be observed that the Maglev system retained its stability for up to three consecutive timeouts (beginning at $t = 1.16$ s) and the sensor voltage remained at 4.5 V. However, right after four consecutive timeouts (beginning at $t = 1.28$ s), the closed-loop system began losing its stability. The two additional consecutive timeouts (beginning at $t = 1.38$ s) did not help the system recover the stability, the steel ball eventually fell down, and the sensor voltage reached the saturated value of 8.5 V (at $t = 1.53$ s). Thus, our timeout and sensor data prediction scheme ensured the Maglev system's stability for sporadic surges in time delay bounded by three sampling periods. We conducted these experiments many times, and the results were very consistent.

V. CONCLUSION

A novel network-based control methodology with delayed/lost sensor data prediction was presented in this paper. Its objective is to accommodate more time delays due to delayed/lost sensor data, maintaining the system stability in the event of sporadic surges in time delay. We developed a client/server architecture for the supervisory control using a CGI/HTML interface with which a client can connect to the host PC remotely via the Internet. We validated the architectures of supervisory control and feedback control over network experimentally using an open-loop unstable ball Maglev system. The proposed real-time control methodology is based on a timeout scheme and sensor data prediction with a fifth-order AR model. This control methodology increased the mean allowable time delay when the time delay followed a uniform distribution. The methodology also ensured the system stability even in the presence of bounded sporadic surges in time delays of three sampling periods (9 ms). This is an almost factor-of-100 (9 ms/90 μs) improvement in accommodating communication time delays. Thus, we experimentally demonstrated the feasibility and effectiveness of the proposed methodology in enhancing the controller's capability of accommodating larger communication delays in network-based real-time control.

REFERENCES

[1] K. Goldberg, S. Gentner, and J. Wiegley, "The Mercury Project: A feasibility study for Internet robots," UC Berkeley and Univ. Southern California, Tech. Rep., 1994.

[2] C. E. Garcia, R. Carelli, J. F. Postigo, and C. Soria, "Supervisory control for a telerobotic system: A hybrid control approach," *Contr. Eng. Practice*, vol. 11, no. 7, pp. 805–817, Jul. 2003.

[3] P. G. Backes, K. S. Tso, and G. K. Tharp, "Mars pathfinder mission internet based operations using WITS," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, May 1998, pp. 284–291.

[4] R. Luck and A. Ray, "An observer-based compensator for distributed delays," *Automatica*, vol. 26, no. 5, pp. 903–908, Sep. 1990.

[5] G. C. Walsh, Y. Hong, and L. G. Bushnell, "Stability analysis of networked control system," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 3, pp. 438–446, May 2002.

[6] R. Krotolica, Ü. Özgüner, H. Chan, H. Göktaş, J. Winkelman, and M. Liubakka, "Stability of linear feedback systems with random communication delays," *Int. J. Contr.*, vol. 59, no. 4, pp. 925–953, Oct. 1994.

[7] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[8] J. Nilsson, "Real time control systems with delays," Ph.D. dissertation, Dep. Automatic Control, Lund Institute of Technol., 1998.

[9] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 4, pp. 503–518, Jul. 2002.

[10] N. J. Ploplys and A. G. Alleyne, "UDP network communications for distributed wireless control," in *Proc. 2003 American Contr. Conf.*, Jun. 2003, pp. 3335–3340.

[11] S. C. Paschall II, "Design, fabrication and control of a single actuator magnetic levitation system," Senior Honors Thesis, Texas A&M University, May 2002.

[12] A. Srivastava and W.-J. Kim, "Internet-based supervisory control and stability analysis for time delay," in *Proc. 2003 American Control Conf.*, Jun. 2003, pp. 627–632.

[13] A. Srivastava, "Distributed real-time control via the Internet," M.S. thesis, Texas A&M Univ., May 2003.

[14] A. Acharya and J. Saltz, "A study of internet round-trip delay," UMIACS and Dep. Computer Science, Univ. Maryland, College Park, MD, Tech. Rep. CS-TR-3736, Dec. 1996.

[15] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Commun. Mag.*, vol. 32, no. 3, pp. 70–81, Mar. 1994.

[16] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: The MIT Press, 1983.