Networked Real-Time Control Strategy Dealing With Stochastic Time Delays and Packet Losses

Won-jong Kim[1]
e-mail: wjkim@tamu.edu

Kun Ji

Ajit Ambike

Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843-3123

A novel model-predictive-control strategy with a timeout scheme and p-step-ahead state estimation is presented in this paper to overcome the adverse influences of stochastic time delays and packet losses encountered in network-based distributed real-time control. An open-loop unstable magnetic-levitation (maglev) test bed was constructed and employed for its experimental verification. The compensation algorithms developed in this paper deal with the network-induced stochastic time delays and packet losses in both the forward path and the feedback path simultaneously. With the p-sampling-period delay upper bound, the networked control system (NCS) can also accommodate up to $p-1$ successive packet losses. Experimental results demonstrate the feasibility and effectiveness of this networked real-time control strategy. [DOI: 10.1115/1.2232692]

Keywords: networked control system, time delays, packet losses, model prediction

1 Introduction

Networked control systems (NCSs) contain sensors, actuators, and estimation and control units connected via communication networks [1–5]. Time delays and packet losses in the communication networks can severely degrade the performance of a real-time control system. The delay characteristics of an NCS primarily depend on the type of the network used. When random-access local-area networks (LANs) such as controller area network (CAN) and Ethernet are used, the time delays usually vary randomly and can be shorter or longer than the sampling period [6].

The communication network in the feedback loop makes the analysis and design of NCSs complicated. The tools and methods developed in the conventional control theory are not effective and should be modified to account for this additional complexity [7,8]. Various methodologies have been proposed based on several types of network behaviors and configurations in conjunction with different ways to treat the delay problem [9–19].

Our research focuses on a general NCS configuration with time delays and packet losses in both the sensor feedback path and the control forward path [2]. A model estimator is used at the controller node with two functions: (1) It recreates the plant dynamics for state estimation in the forward loop; and (2) it acts as an autoregressive (AR) prediction model with a timeout scheme for delayed/lost sensor data in the feedback loop. The main contribution of this paper is that we developed and experimentally verified a networked real-time control strategy to deal with network-induced time delays and packet losses in both the forward path and the feedback path in a unified way. We constructed an open-loop unstable maglev system as a test bed and connected this to an existing LAN for the purpose of the experimental verification of the compensation algorithms. We experimentally observed delay in our test bed, also introduced additional delay in experiments, and verified the effectiveness of the proposed networked real-time control strategy.

2 Plant Model With Stochastic Time Delays

In this paper, we assume the network-induced time delays are randomly varying with unknown distributions, however, the upper bound of the time delay is less than p sampling intervals or the number of consecutive packet losses is less than $p-1$. Consider a continuous-time single-input single-output (SISO) system, where the dynamics is represented as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned} \tag{1}$$

where $x(t) \in R^n$, $u(t) \in R$, $y(t) \in R$, and A, B, and C are constant matrices of compatible dimensions, and $v(t)$ and $w(t)$ are uncorrelated white Gaussian noises. The discrete-time model of the system depends on the length of delay. Denote l sampling intervals as the bound of the varying length of the time delay at a given instance of time and $l=1,2,3,\ldots$, or p. Sampling with the sampling period h gives the discrete-time NCS with the round-trip time delay τ_n as follows:

$$\begin{aligned} x(n+1) &= \Phi x(n) + \Gamma U(N-1) + v(n) \\ &\cdots \\ x(n+l-1) &= \Phi x(n+l-2) + \Gamma U(N-1) + v(n+l-2) \\ x(n+l) &= \Phi x(n+l-1) + \Gamma_0(\tau_n)U(N) + \Gamma_1(\tau_n)U(N-1) + v(n+l-1) \\ y(n) &= Cx(n) + w(n) \end{aligned} \tag{2}$$

where $\Phi=e^{Ah}$, $\Gamma_0(\tau_n)=\int_0^{lh-\tau_n}e^{As}Bds$, $\Gamma_1(\tau_n)=\int_{lh-\tau_n}^{h}e^{As}Bds$, $\Gamma=\int_0^h e^{As}Bds$, and $\tau_n<lh$. The time-driven index n is the index of the sampling-time instant. The event-driven index N is the index of control packet arrival. The relation between $U(N)$ and $u(n)$ will be established in Sec. 3.2.

3 Compensation for Network-Induced Time Delays and Packet Losses

Two classes of time delays included in an NCS are shown in Fig. 1: (1) The delay τ_{sc} from the sensor node to the controller node and (2) the delay τ_{ca} from the controller node to the actuator node. τ_{sc} and τ_{ca} are different in nature and two separate algorithms are needed to deal with them independently.

3.1 AR Model With a Timeout Scheme for τ_{sc} and Packet-Loss Compensation. In Fig. 1, the controller node is modeled as

$$\begin{aligned} y(n) &= Cx(n) + w(n) \\ u(n) &= -L_y y(n - m(\tau_{sc}(n))) \end{aligned} \tag{3}$$

where $\tau_{sc}(n)$ denotes the sensor-to-controller time delay at the nth sampling interval. The output feedback controller is represented with a gain matrix L_y. The finite non-negative integer $m(\tau_{sc}(n))$ represents the number of delayed/lost sensor data in the sensor-to-controller path, defined as

[1]Corresponding author.

Contributed by the Dynamic Systems, Measurement, and Control Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received August 18, 2004; final manuscript received June 21, 2005. Assoc. Editor: George Chiu.

Copyright © 2006 by ASME

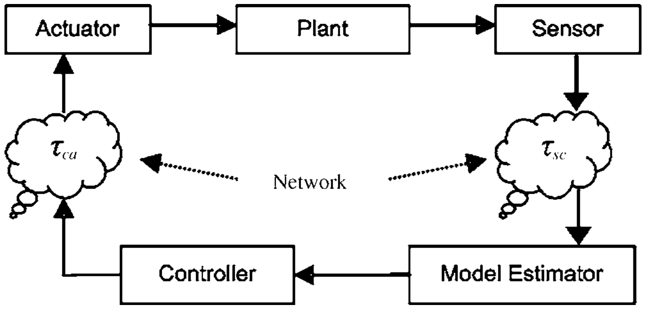

Fig. 1 Distributed real-time control system with network-induced time delays

$$m(\tau_{sc}(n)) = \left\lfloor \frac{h + \tau_{sc}(n) - \tau_0}{h} \right\rfloor \quad (4)$$

where τ_0 represents the timeout threshold, and the symbol $\lfloor . \rfloor$, a truncation function. If the latest sensor measurement is not available for a preset threshold time τ_0 in each sampling period, the controller gives up and uses an estimated value generated by an AR model [20].

$$\hat{y}(n) = -a_1 y(n-1) - \cdots - a_k y(n-k) \quad (5)$$

where k is a non-negative integer. Since $m(\tau_{sc}(n))$ varies from one timeout case to another, we extend this methodology to the cases when multiple consecutive timeouts occur [21].

3.2 Prediction Algorithm for τ_{ca} and Packet-Loss Compensation. The controller does not know how long it will take for its control signal to reach the actuator. However, by assumption, the worst case is that the control signal does not arrive at the actuator node in p sampling intervals. Thus we design an estimator at the controller node to estimate the plant states of the successive samples p steps in advance. With these estimated states the controller calculates the control signals for each of the following p sampling intervals and sends them as a package to the actuator node. The actuator node then adopts the corresponding control signal from the package in the current sampling interval. In case the new control-signal package does not arrive, the actuator node can use formerly calculated and stored control signals for up to the next $p-1$ sampling intervals from the control signal package that arrived most recently.

The p-step-ahead state estimation is done as follows:

$$\begin{aligned}\hat{x}(n+1) &= \Phi x(n) + \Gamma u(n)\\ \hat{x}(n+2) &= \Phi \hat{x}(n+1) + \Gamma u(n+1)\\ &\cdots\\ \hat{x}(n+p) &= \Phi \hat{x}(n+p-1) + \Gamma u(n+p-1).\end{aligned} \quad (6)$$

The control signal package is generated as

$$\begin{aligned}u(n) &= -Lx(n)\\ u(n+1) &= -L\hat{x}(n+1)\\ &\cdots\\ u(n+p-1) &= -L\hat{x}(n+p-1),\end{aligned} \quad (7)$$

where L denotes the control law without assuming delay and is determined in the controller design section (Sec. 4). Thus each control signal package transmitted to the actuator node includes $u(n), u(n+1), \ldots, u(n+p-1)$. The actuator node chooses the control signal $U(N)$ from the package as below for the next p sampling intervals until the new control signal package arrives.

$$\begin{aligned}&\text{if } \tau_{ca}(n) \leqslant h, \text{ then}\\ &U(N) = u(n) \text{ for } nh \leqslant t < (n+1)h\\ &U(N+1) = u(n+1) \text{ for } (n+1)h \leqslant t < (n+2)h\\ &\cdots\\ &\textit{if } h \leqslant \tau_{ca}(n) < 2h, \text{ then}\\ &U(N) = u(n) \text{ for } nh \leqslant t < (n+1)h\\ &U(N) = u(n+1) \text{ for } (n+1)h \leqslant t < (n+2)h\\ &U(N+1) = u(n+2) \text{ for } (n+2)h \leqslant t < (n+3)h\\ &\cdots\\ &\text{if } (p-1)h \leqslant \tau_{ca}(n) < ph, \text{ then}\\ &U(N) = u(n) \text{ for } nh \leqslant t < (n+1)h\\ &\cdots\\ &U(N) = u(n+p-1) \text{ for } (n+p-1)h \leqslant t < (n+p)h\\ &U(N+1) = u(n+p) \text{ for } (n+p)h \leqslant t < (n+p+1)h\\ &\ldots,\end{aligned} \quad (8)$$

where t denotes the continuous time. Note that $U(N)$ is unchanged during the controller-to-actuator time delay τ_{ca} and then updated to be $U(N+1)$ using the incoming new control signal package.

For example, with $p=4$, the control signals adopted by the actuator corresponding to the sampling intervals are illustrated in Fig. 2 when different time delays or packet losses occur. Note again that N is the event-driven index whereas n is the time-driven index.

If there is no new control signal package available in any given sampling interval, the formerly estimated control signal by (8) and stored in the last available control signal package is used. If a new control signal package arrives in any given sampling interval, it revises all the components of the last control signal package. Then $U(N+1)$ is now available. From Fig. 2 we can see that the actuator gets the corresponding control signal $u(n)$ in each sampling interval regardless of time delays or packet losses. The effect of packet losses is no worse than that of time delays. Thus this strategy works when there are both time delays and packet losses. In case of an out-of-order transmission and arrival of packets, the outdated packets are simply discarded. Thus our strategy also deals with out-of-order packet transmissions.

3.3 Determination of p in Real-Time NCS. The parameter p depends on the characteristics of the NCS and the accuracy of the plant model for state estimation. In practical implementation of our time-delay or packet-loss compensation strategy presented in this section, some key NCS characteristics, such as delay and packet-loss characteristics and bound, network sampling frequency and bandwidth, etc. must be determined a priori, and then the minimum p can be determined. Assuming a larger p can maintain system stability in the presence of longer time delays. However, it will lead to excessive computation time, and the size of the control signal package will increase. This will use up more communication bandwidth and cause longer time delay. Thus there is an engineering trade-off between the value of p and the overall NCS performance. An example of choosing p is shown in Sec. 5 with our specific test bed.

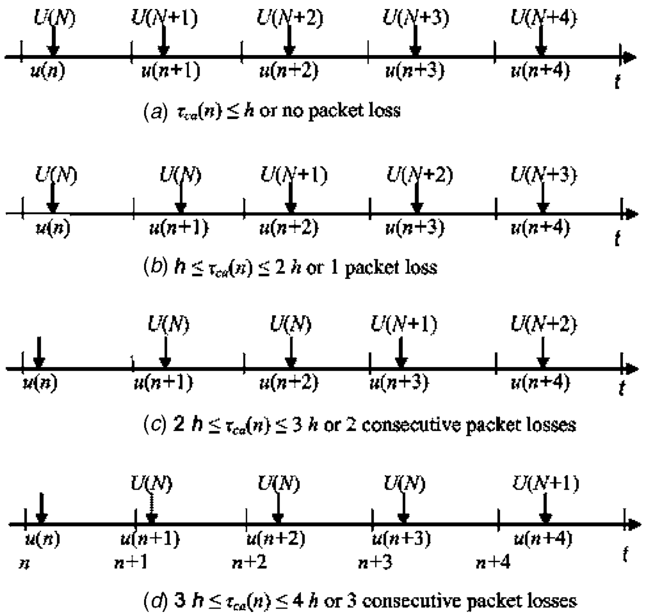

Fig. 2 Actual control signal $U(N)$ adopted by the actuator node and components of the control signal packet $u(n)$ with respect to time when different time delays or packet losses occur. Figures (a)–(d) cover all the possible cases of the time delay $\tau_{ca}(n)$ up to 4 h or the number of consecutive packet losses up to 3. Short vertical lines indicate sampling instants. Solid arrows indicate that new control signal packages arrive in the corresponding sampling interval. Dotted arrows indicate that the incoming control signal packages are delayed or lost.

4 Controller Design

Along with the compensation algorithms described in Sec. 3, we develop two approaches to design controllers to control the system with network-induced time delays and packet losses. One approach is to design a classical controller without taking the time delays and the packet losses into consideration. The other approach is to design an optimal controller with the consideration of the time delays and packet losses [22]. We present the optimal controller design in this section.

Introduce the estimation error $\tilde{x}(n)=x(n)-\hat{x}(n)$, and consider the following two cases.

1. An updated control signal package is available. When the new control signal is available, the state equation is $x(n+1)=\Phi x(n)+\Gamma_0(\tau_n)U(N)+\Gamma_1(\tau_n)U(N-1)+v(n)$ from (2) and the estimation equation is $\hat{x}(n+1)=\Phi x(n)+\Gamma u(n)$ from (6). $U(N)$ will replace $U(N-1)$. Thus from (8) the system state equation is updated as

$$\begin{aligned} x(n+1) &= \Phi x(n)+\Gamma_0(\tau_n)u(n)+\Gamma_1(\tau_n)u(n)+v(n) \\ &= \Phi x(n)+\Gamma u(n)+v(n). \end{aligned} \tag{9}$$

2. No updated control signal package is available. When no new control signal is available, the state equation is $x(n+1)=\Phi x(n)+\Gamma U(N-1)+v(n)$ from (2), and the estimation equation is $\hat{x}(n+1)=\Phi x(n)+\Gamma u(n)$ from (7). From (8) we have $U(N-1)=u(n)$ in the nth sampling interval, and the system equation is also in the form of (9).

For both the two cases, we have the error dynamics

$$\tilde{x}(n+1)=\Phi\tilde{x}(n)+v(n) \tag{10}$$

From (6)–(10) the augmented closed-loop system can be written as

$$z(n+1)=\underline{\Phi}z(n)+\underline{\Gamma}e(n) \tag{11}$$

where

$$z(n)=\begin{bmatrix} x(n) \\ \tilde{x}(n) \\ u(n-1)\end{bmatrix}, \quad \underline{\Phi}=\begin{bmatrix} \Phi & 0 & \Gamma \\ 0 & \Phi & 0 \\ -L & L & 0\end{bmatrix},$$

$$e(n)=\begin{bmatrix} v(n) \\ w(n)\end{bmatrix}, \quad \underline{\Gamma}=\begin{bmatrix} I & 0 \\ I & 0 \\ I & L\end{bmatrix}$$

and the covariance $R_{ee}=E\{e(n)e(n)^T\}$. Denote the state covariance $P_n=E\{z(n)z(n)^T\}$, then

$$P_{n+1}=E\{z(n+1)z(n+1)^T\}=E\{\underline{\Phi}P_n\underline{\Phi}^T+\underline{\Gamma}R_{ee}\underline{\Gamma}^T\} \tag{12}$$

Then the quadratic cost function is $J=E\{z(n)^TQz(n)\}$ with

$$\lim_{n\to\infty}E\{z(n)^TQz(n)\}=E\{QP_\infty\} \tag{13}$$

where $P_\infty=\lim_{n\to\infty}P_n$. The cost function can also be written as

$$J=E\{x(n)^TQ_xx(n)+u(n)^TQ_uu(n)\} \tag{14}$$

where Q is related with Q_x and Q_u as

$$Q=\begin{bmatrix} Q_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix} -L^T \\ L^T \\ 0\end{bmatrix}Q_u[-L\ L\ 0] \tag{15}$$

The control law for the optimal state feedback is derived by using dynamic programming and is described as [23,24]

$$u(n)=-L\hat{x}(n), L_yC=L \tag{16}$$

where $L=-(Q_u+\Gamma^TK\Gamma)^{-1}\Gamma^TK\Phi$ and the matrix K is the unique positive semidefinite solution of the algebraic Riccati equation [23,24]

$$K=\Phi^T(K-K\Gamma(Q_u+\Gamma^TK\Gamma)^{-1}\Gamma^TK)\Phi+Q_x \tag{17}$$

5 Experimental Results

To verify the effectiveness of the algorithms developed in Sec. 3, two sets of experiments were conducted with a network-controlled single-actuator ball maglev system described in [25]. It is an open-loop unstable system, and the stability is lost when actuation of the control signal misses the deadline of 3 ms after sampling. Thus this setup is useful to verify the effectiveness of our compensation algorithms. The client-side setup consists of an actuator, plant, and sensor, and the server-side setup consists of a controller and estimator. The client and the server communicate using unblocked User Datagram Protocol (UDP) sockets over a 100 Mbps LAN. The operating system is Linux with real time application interface (RTAI) [26].

The network-induced time delays were measured to determine the key statistical characteristics of our NCS. In our lab, the average round-trip time delay induced by the LAN is 230 μs and its standard deviation is 200 μs. We observed that there exist sporadic time delays of the order of 4 ms. Because the sampling period of our NCS with the maglev test bed is 3 ms, the network-induced sporadic time delay in our lab is about one sampling period, which is not long enough to verify the effectiveness of the proposed compensation strategies developed in Sec. 3. Thus we introduced longer artificial time delays/packet losses in both the feedback path and forward path for the purpose of demonstrating the effectiveness of our strategies. For our NCS, considering the system sampling frequency and network bandwidth, we chose the parameter $p=5$. That is, we assume the time-delay upper bound is 15 ms or the upper bound of the number of successive packet losses is 4.

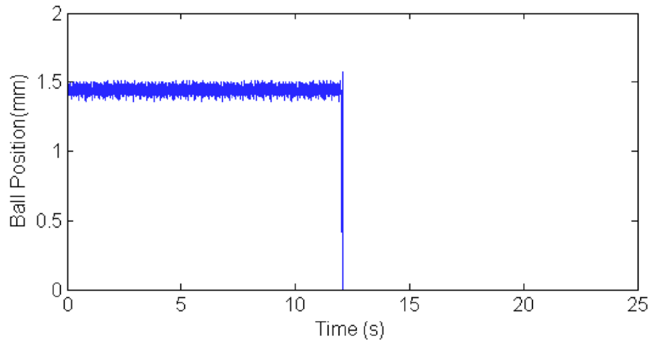

Fig. 3 Ball position with packets losses beginning at $t=12$ s

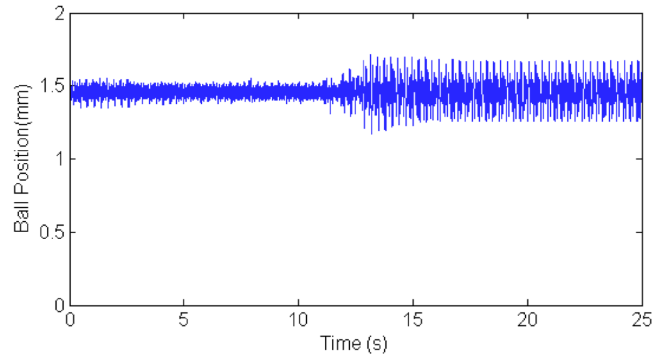

Fig. 5 Ball position with 20% packet losses beginning at $t=12$ s onwards

5.1 Ball Position Regulation With and Without Time-Delay/Packet-Loss Compensation. The first set of experiments was conducted to verify the effectiveness of the algorithms. First, no time-delay/packet loss compensation was used. At $t=12$ s, we forced one data packet to be lost while transmitted from the sensor node to the controller node and from the controller node to the actuator node. The response of the system is shown in Fig. 3. In this figure, the zero value of the vertical axis denotes that the system lost its stability and that the ball could not maintain its equilibrium position and fell down due to the absence of the proper control signal.

After performing many design iterations with various models of different orders, the following fifth-order AR model was chosen as the sensor data prediction and was sufficiently accurate with minimal execution time in the control loop [21].

$$\hat{y}(n) = 0.3195y(n-1) + 0.2669y(n-2) - 0.0622y(n-3) + 0.1960y(n-4) + 0.4064y(n-5). \tag{18}$$

The time-delay/packet-loss compensation algorithm (8) was also implemented to compensate for the packet loss from the controller node to the actuator node. In the second experiment, artificial packet losses were introduced and four successive packet losses occurred every 6 s from $t=3$ s onwards. The system response is shown in Fig. 4. The NCS test setup maintained its stability successfully with periodic 1.1 mm peak-to-peak spikes in the ball position.

In the third experiment, the compensation algorithms were also implemented. Artificial packet losses were introduced every fifth sample (i.e., at the 20% packet-loss rate) after $t=12$ s. The system response is shown in Fig. 5, and the system remained stable throughout the experiment. The fluctuation in the ball position about the equilibrium point increased by a fact of 2.5 after the 20% packet losses were introduced. The performance was degraded due to the multi-step-ahead state estimation that would inevitably introduce estimation errors.

5.2 Ball-Position Step Responses and Dynamic Tracking With and Without Time-Delay/Packet-Loss Compensation. The second set of experiments was conducted to determine the degradation of system performance due to time delays or packet losses in step response and dynamic tracking. Figure 6 shows a closed-loop step (started at $t=7$ s) response of the test setup without packet loss. Figure 7 shows the step (started at $t=7$ s) response with packet losses. The average packet loss rate is 20%. The closed-loop system was stable with a worse stability margin due to packet loss and imperfect state estimation.

Figure 8 shows the compensated system response of tracking a sinusoidal position command at the average packet-loss rate of 20% from $t=50$ s onwards. Repeating experiments with various frequencies, we concluded that the closed-loop system bandwidth was reduced from 2.7 to 0.34 Hz due to packet losses and imperfect state estimation.

Figures 9 and 10 show the compensated system response of tracking a saw-tooth position command at different packet-loss rates from $t=34$ s onwards. Comparing Figs. 9 and 10, we can see that the fluctuation in the ball position increases with the increase of the packet-loss rate.

6 Conclusions

In this paper, we presented a networked real-time control strategy to deal with stochastic time delays and packet losses in NCSs. Novel model-based estimation algorithms were developed to compensate for the two classes of time delays and packet losses simultaneously. We constructed and employed a maglev test bed over a 100 Mbps Ethernet using unblocked UDP sockets for ex-

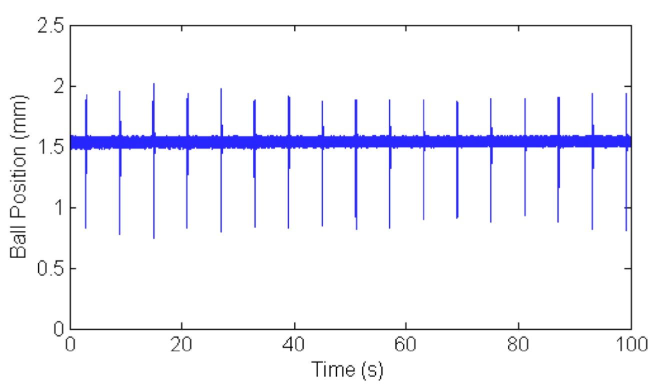

Fig. 4 Ball position with four successive packet losses occurring every 6 s

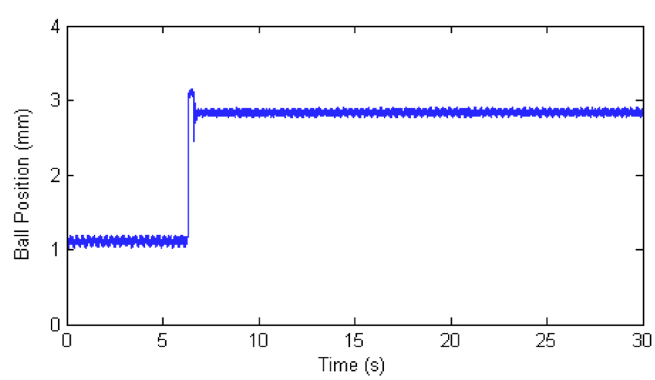

Fig. 6 Step response without packet loss

perimental verification. The experimental results demonstrated the feasibility and effectiveness of these control algorithms. We could stabilize our open-loop-unstable maglev test bed even with time delays/packet losses up to five sampling periods at a packet-loss rate up to 20%. The tracking capability of the closed-loop maglev system was also retained with degraded noise performance due to packet losses and imperfect state estimation.

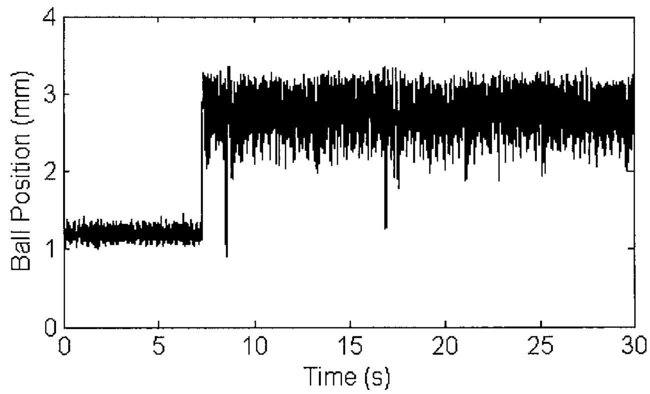

Fig. 7 Step response with 20% packet losses

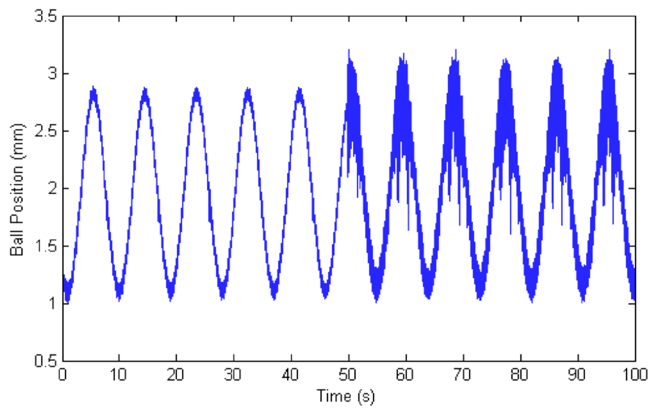

Fig. 8 Tracking a sinusoidal command with packet losses beginning at t=50 s

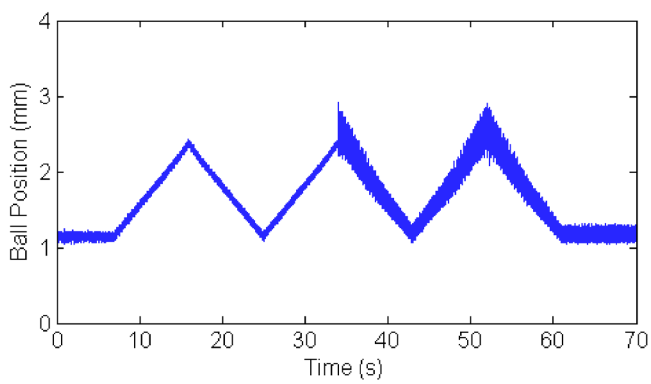

Fig. 9 System tracking response with 10% packet losses

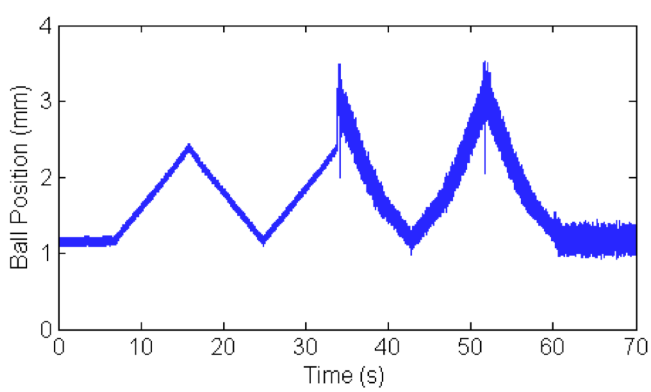

Fig. 10 System tracking response with 20% packet losses

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CMS-0116642. We thank Abhinav Srivastava and Steven Paschall for their contributions.

References

[1] Chan, H., and Ozguner, U., 1995, "Closed-Loop Control of Systems Over a Communications Network With Queues," Int. J. Control, **62**(3), pp. 493–510.
[2] Tipsuwan, Y., and Chow, M.-Y., 2003, "Control Methodologies in Networked Control Systems," Control Eng. Pract., **11**(10), pp. 1099–1111.
[3] Schickhuber, G., and McCarthy, O., 1997, "Distributed Fieldbus and Control Network Systems," Comput. Control Eng. J., **8**(1), pp. 21–32.
[4] Raji, R. S., 1994, "Smart Networks for Control," IEEE Spectrum, **31**(6), pp. 49–55.
[5] Elia, N., and Mitter, S., 2001, "Stabilization of Linear Systems With Limited Information," IEEE Trans. Autom. Control, **46**(9), pp. 1384–1400.
[6] Lian, F.-L., Moyne, J. R., and Tilbury, D. M., 2001, "Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet," IEEE Control Syst. Mag., **21**(1), pp. 66–83.
[7] Walsh, G. C., Hong, Y., and Bushnell, L. G., 2002, "Stability Analysis of Networked Control System," IEEE Trans. Control Syst. Technol., **10**(3), pp. 438–446.
[8] Zhang, W., Branicky, M. S., and Phillips, S. M., 2001, "Stability of Networked Control Systems," IEEE Control Syst. Mag., **21**(1), pp. 84–99.
[9] Halevi, Y., and Ray, A., 1988, "Integrated Communication and Control Systems: Part I—Analysis," ASME J. Dyn. Syst., Meas., Control, **110**(4), pp. 367–373.
[10] Halevi, Y., and Ray, A., 1988, "Integrated Communication and Control Systems: Part II—Design Consideration," ASME J. Dyn. Syst., Meas., Control, **110**(4), pp. 374–381.
[11] Luck, R., and Ray, A., 1990, "An Observer-Based Compensator for Distributed Delays," Automatica, **26**(5), pp. 903–908.
[12] Walsh, G. C., Beldiman, O., and Bushnell, L., 1999, "Asymptotic Behavior of Networked Control Systems," *Proceedings of the IEEE International Conference on Control Applications*, Kohala Coast, HI, 2(3), pp. 1448–1453.
[13] Krotolica, R., Özgüner, U., Chan, H., Göktaş, H., Winkelman, J., and Liubakka, M., 1994, "Stability of Linear Feedback Systems with Random Communication Delays," Int. J. Control, **59**(4), pp. 925–953.
[14] Liou, L. W., and Ray, A., 1991, "A Stochastic Regulator for Integrated Communication and Control Systems: Part I—Formulation of Control Law," ASME J. Dyn. Syst., Meas., Control, **113**(4), pp. 604–611.
[15] Nilsson, J., 1998, *Real-Time Control Systems with Delays*, Ph. D. dissertation, Dept. Automatic Control. Lund Institute of Technology, Lund, Sweden.
[16] Lian, F.-L., Moyne, J. R., and Tilbury, D. M., 2001, "*Networked Control Systems Toolkit: A Simulation Package for Analysis and Design of Control Systems with Network Communication*," Technical Report, UM-ME-01–04, Department of Mechanical Engineering, University of Michigan, Ann Arbor.
[17] Ploplys, N. J., Kawka, P. A., and Alleyne, A. G., 2004, "Closed-Loop Control Over Wireless Network," IEEE Control Syst. Mag., **24**(3), pp. 58–71.
[18] Almutairi, N. B., Chow, M.-Y., and Tipsuwan, Y., 2001, "Network-Based Controlled DC Motor With Fuzzy Compensation," *Proc. of the 27th Annual Conference of the IEEE Industrial Electronics Society*, **3**(5), pp. 1844–1849.
[19] Yook, J. K., Tilbury, D. M., and Soparkar, N. R., 2002, "Trading Computation for Bandwidth: Reducing Communication in Distributed Control System Using State Estimators," IEEE Trans. Control Syst. Technol., **10**(4), pp. 503–518.
[20] Ljung, L., and Söderström, T., 1983, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Massachusetts.
[21] Srivastava, A., 2003, Distributed Real-Time Control via the Internet, MS thesis, Texas A&M University.
[22] Lian, F. -L., and Moyne, J. R., 2003, "Modelling and Optimal Controller Design of Networked Control Systems With Multiple Delays," Int. J. Control, **76**(6), pp. 591–606.
[23] Bertsekas, D. P., 1995, *Dynamic Programming and Optimal Control*, Athena Scientific, Blemont, MA, Vol. 1.
[24] Lewis, F., 1986, *Optimal Estimation*, Wiley, New York.
[25] Paschall, S. C., II, 2002, "Design, Fabrication and Control of a Single Actuator Magnetic Levitation System," Senior Honors thesis, Texas A&M University.
[26] http://www.rtai.org.