Networked Real-Time Control Strategies Dealing with Stochastic Time Delays and Packet Losses

Won-jong Kim, Kun Ji, and Ajit Ambike

Abstract—**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 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.**

Keywords—**Networked control system, time delays, packet losses, model prediction.**

I. Introduction

Networked control systems (NCSs) contain sensors, actuators, and estimation and control units connected via communication networks [1−4]. Time delays and packet losses in the communication network can severely degrade the performance of real-time control system. Network-induced time delays originate from time sharing of communication media as well as additional functionalities required for physical signal coding and processing. When there is congestion in the communication network, some data packets are dropped to either reduce the queue size or to inform the senders to lower their transmission rates. The delay characteristics of an NCS primarily depend on the type of network used. When random-access local-area networks 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 [5–7].

Various methodologies have been proposed based on several types of network behaviors and configurations in conjunction with different ways to treat the delay problem [8–18]. Ray *et al*. [8–9] formulated an augmented discrete-time model methodology to control a linear plant over a periodic delay network. A queuing methodology developed by Luck and Ray [10] used an observer to estimate the plant states and a predictor to compute the predictive control based on past output measurements. Walsh *et al*. [11] used a nonlinear and perturbation theory to compensate for network delay effects in an NCS as the vanishing perturbation of a continuous-time system under the assumption that there is no observation noise. Krotolica *et al*. [12] designed a networked controller in the frequency domain using robust control theory. Liou and Ray modeled the communication delay as stochastic but bounded, and developed a linear quadratic optimal controller [13]. Nilsson [14] proposed an optimal stochastic control methodology to control an NCS on random delay networks. To analyze and design an NCS, a toolkit was developed by Lian *et al*. [15]. Ploplys and Alleyne developed a distributed-control system for a Furuta pendulum over a dedicated wireless communication network [16]. Almutairi *et al*. [17] proposed a fuzzy logic modulation methodology for an NCS based on the fuzzy-logic theory. Yook *et al*. [18] proposed a framework for distributed control systems in which estimators are used at each node.

Our research focuses on a general and practical NCS configuration with time delays and packet losses in both the sensor feedback path and the control input path [2]. In this paper we propose novel model-based prediction and compensation algorithms to compensate for network-induced stochastic time delays and packet losses. A plant model is used at the controller node with two functions: (1) it re-creates the plant dynamics for state prediction in the forward loop; (2) it acts as an autoregressive (AR) prediction model with a timeout scheme for delayed/lost sensor data in the feedback loop. The unavailable states are estimated several steps in advance with the plant model. Then the control strategies compensate for time delays and packet losses in both the forward and feedback paths at the same time.

In Section II we present the problem statement with a discrete-time plant model with stochastic time delays. In Section III control strategies for time-delay and packet-loss compensation are presented. Section IV describes the controller design. Experimental results that verify these control strategies are presented in Sections V.

II. Plant Model with Stochastic Time Delays

In this paper, the following assumptions are made:

1. The sensor node and the controller node are time-driven, and the actuator node is both time-driven (in

This material is based upon work supported by the National Science Foundation under Grant No. CMS-0116642.

W. -J. Kim is with Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123, USA (e-mail: wjkim@mengr.tamu.edu). Phone: (979)845-3645, Fax: (979)862-3989.

K. Ji is with Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123, USA (e-mail: kunji@tamu.edu).

A. Ambike is currently a Software Engineer at NuView, Inc. in Houston, TX, USA (e-mail: ajitambike@yahoo.com).

case of signal overdue) and event-driven.

2. The network-induced time delay is randomly varying and the upper bound of the total time delay is less than p sampling intervals and the number of consecutive packet losses is less than $p - 1$.

The dynamics of the continuous-time plant is

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

where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^r$, 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, ⋯, or p. Sampling with the sampling period h gives the discrete-time NCS as follows

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

where

$$\Phi = e^{Ah}, \Gamma_0(\tau_n) = \int_0^{lh-\tau_n} e^{As} B ds, \Gamma_1(\tau_n) = \int_{lh-\tau_n}^{h} e^{As} B ds, \Gamma = \int_0^{h} e^{As} B ds$$

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 Section III.B.

III. 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 to the controller, and (2) the delay τ_{ca} from the controller to the actuator. The delay τ_{sc} and τ_{ca} are different in nature. Thus we need two different algorithms to deal with these two classes of time delays independently.

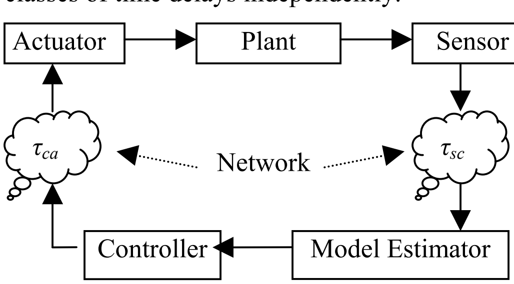

Fig. 1. Distributed real-time control system with network-induced time delays.

A. 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 n-th 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 and it can be defined as

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

where τ_0 represents the timeout threshold, and the symbol $\lfloor . \rfloor$ represents a truncation function. A timeout scheme is used here; if the latest sensor measurement is not available for a preset threshold time τ_0 in each sampling period, the controller gives up for the availability of the actual sensor measurement and uses an estimated value which is generated by the model estimator in Fig. 1.

An AR model is used in this paper for the sensor data estimation $\hat{y}$ because of its simplicity in implementation [19]. Since $m(\tau_{sc}(n))$ varies from one timeout case to another, we can extend this methodology to the cases when multiple consecutive timeouts occur.

B. Prediction Algorithm for τ_{ca} and Packet-Loss Compensation

Since the controller does not know how long it will take for its control signal to reach the actuator, no correction can be made at the controller node. However, the worst case is that the control signal does not arrive at the actuator node in p sampling intervals by Assumption 2. This could be due to the time delay that is up to p sampling-period long or p – 1 consecutive packet losses between the controller node and the actuator node. 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 then sends them as a package to the actuator node. The actuator node 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 control signals for up to the next p – 1 sampling intervals from the package that arrived most recently.

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

$$\begin{aligned} \hat{x}(n+1) &= \Phi \hat{x}(n) + \Gamma u(n) \\ \hat{x}(n+2) &= \Phi \hat{x}(n+1) + \Gamma u(n+1) \\ &\ldots \\ \hat{x}(n+p) &= \Phi \hat{x}(n+p-1) + \Gamma u(n+p-1) \ . \end{aligned} \tag{5}$$

The control signal package is generated as

$$\begin{aligned} u(n) &= -L\hat{x}(n) \\ u(n+1) &= -L\hat{x}(n+1) \\ &\ldots \\ u(n+p-1) &= -L\hat{x}(n+p-1) \ , \end{aligned} \tag{6}$$

where L denotes the control law without assuming delay and is determined in Section IV. Thus each control signal package transmitted to the actuator node includes $u(n)$, $u(n + 1)$,…,and $u(n + p - 1)$. 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

$$\begin{aligned}
&if \quad \tau_{ca}(n) \le h, \quad then \\
&U(N) = u(n) \quad for \quad nh \le t < (n+1)h; \\
&U(N+1) = u(n+1) \quad for \quad (n+1)h \le t < (n+2)h; \\
&\ldots \\
&if \quad h \le \tau_{ca}(n) < 2h, \quad then \\
&U(N) = u(n) \quad for \quad nh \le t < (n+1)h; \\
&U(N) = u(n+1) \quad for \quad (n+1)h \le t < (n+2)h; \\
&U(N+1) = u(n+2) \quad for \quad (n+2)h \le t < (n+3)h; \\
&\ldots \\
&if \quad (p-1)h \le \tau_{ca}(n) < ph, \quad then \\
&U(N) = u(n) \quad for \quad nh \le t < (n+1)h; \\
&\ldots \\
&U(N) = u(n+p-1) \quad for \quad (n+p-1)h \le t < (n+p)h \\
&U(N+1) = u(n+p) \quad for \quad (n+p)h \le t < (n+p+1)h \\
&\ldots
\end{aligned} \quad , \tag{7}$$

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 index while n is the discrete-time index.

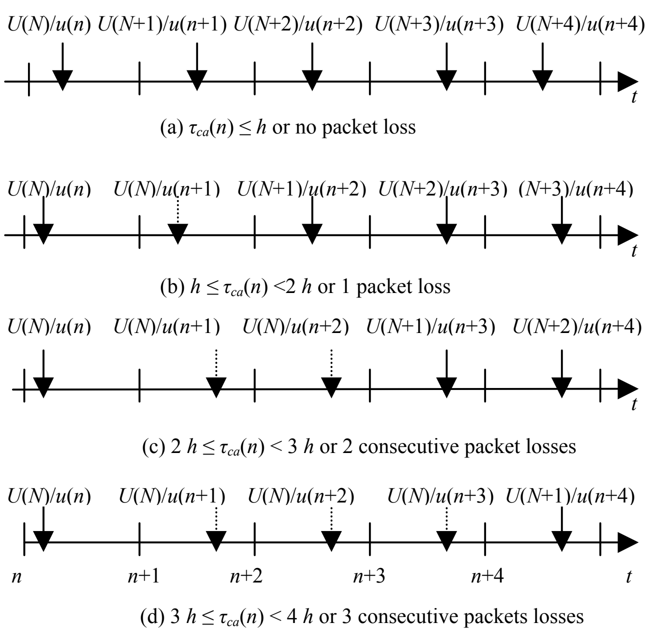

Fig. 2. Actual control signal $U(N)$ adopted by the actuator node and components of the control signal packet $u(n)$ with respect to time when different time delays or packet losses occur. In this figure p is assumed to be 4, and Parts (a), (b), (c) and (d) cover all the possible cases of the time delay $\tau_{ca}(n)$ up to 4 h or the number of consecutive packet losses up to 3. Short vertical lines indicate sampling instants. Solid arrows indicate that a new control signal package arrives in the corresponding sampling interval, and dotted, the delayed or lost incoming control signal package.

If there is no new control signal package available in any given sampling interval, the formerly estimated control signal by (7) 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 transition and arrival of packages, the outdated packages are simply discarded.

C. *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. Thus key NCS characteristics must be determined a priori, and then the minimum p can be determined. A larger p may maintain system stability in the present of longer time delays. However it will lead to excessive computation time and the size of 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. A practical way to determine p is delineated in Section V with our test bed.

IV. Controller Design

Along with the compensation algorithms described in Section III, we develop two approaches to design controllers for a system with time delays and packet losses: (1) a classical controller without taking the time delays and the packet losses into consideration, and (2) 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 (5). $U(N)$ will replace $U(N - 1)$. Thus from (7) the system state equation is updated as

$$\begin{aligned} x(n+1) &= \Phi x(n) + \Gamma_0(\tau_n)u(n) + \Gamma_1(\tau_n)u(n) + v(n) \\ &= \Phi x(n) + \Gamma u(n) + v(n). \end{aligned} \tag{8}$$

2. *No Updated Control Signal Package Is Available*. When no new control signal package 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 (5). From (7), we have $U(N - 1) = u(n)$ in the n-th sampling interval, and the system equation is also in the form of (8).

For both the two cases, we have the error dynamics

$$\tilde{x}(n+1) = \Phi \tilde{x}(n) + v(n) \; . \tag{9}$$

From (5–9) the augmented closed-loop system is

$$z(n+1) = \underline{\Phi} z(n) + \underline{\Gamma} e(n)\,, \tag{10}$$

where

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

and the covariance $R_{ee} = E\{e(n)e(n)^T\}$.

Denote the state covariance $P_n = E\{z(n)z(n)^T\}$, then

$$P_{n+1} = E\{z(n+1)z(n+1)^T\} = E\{\underline{\Phi} P_n \underline{\Phi}^T + \underline{\Gamma} R_{ee} \underline{\Gamma}^T\}. \tag{11}$$

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

$$\lim_{n\to\infty} E\{z(n)^T Qz(n)\} = E\{QP_\infty\}\,, \tag{12}$$

where $P_\infty = \lim_{n\to\infty} P_n$.

Since $w(n)$ is of zero mean, the cost function can also be written as

$$J = E\{x(n)^T Q_x x(n) + u(n)^T Q_u u(n)\}, \tag{13}$$

where Q in (12) is related with Q_x and Q_u as

$$Q = \begin{bmatrix} Q_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -L^T \\ L^T \\ 0 \end{bmatrix} Q_u \begin{bmatrix} -L & L & 0 \end{bmatrix}. \tag{14}$$

The control law for the optimal feedback is derived by using dynamic programming and is described as [21–22]

$$u(n) = -L\hat{x}(n), L_y C = L, \tag{15}$$

where $L = -(Q_u + \Gamma^T K \Gamma)^{-1} \Gamma^T K \Phi$ and matrix K is the unique positive semidefinite solution of the algebraic Riccati equation [21–22]

$$K = \Phi^T (K - K\Gamma (Q_u + \Gamma^T K \Gamma)^{-1} \Gamma^T K) \Phi + Q_x\,. \tag{16}$$

V. Experimental Results

To verify the effectiveness of the algorithms developed in Section 3, two sets of experiments were conducted with an NCS as shown in Fig. 1. Stephen C. Paschall, II, developed the single-actuator magnetic ball levitation system as his senior honors project [23]. 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 local area network. For all the experimental results presented in this section, the control loop was closed over this network. The operating system is Linux with Real Time Application Interface (RTAI) [24]. A digital lead-lag controller was implemented in the server-side PC. For our setup, the network induced time delay was measured to determinate the key characteristics of our NCS. Fig. 3 shows the specification of the round-trip time delays between the client PC and the sever PC. The average round-trip time delay is about 230 μs and the standard deviation is about 200 μs. The sampling period of our ball maglev test bed is 3 ms. Thus we introduced artificial time-delay/packet-loss for the purpose of demonstrating the effectiveness of our strategies developed in Section III. For our NCS, considering the system sampling frequency and network bandwidth, we chose the parameter p = 5, that is, we assume the upper bound of time delays is 15 ms or the upper bound of the number of successive packet losses is 4. This is a good conservative upper bound even with the sporadic surges in time delays shown in Fig. 3.

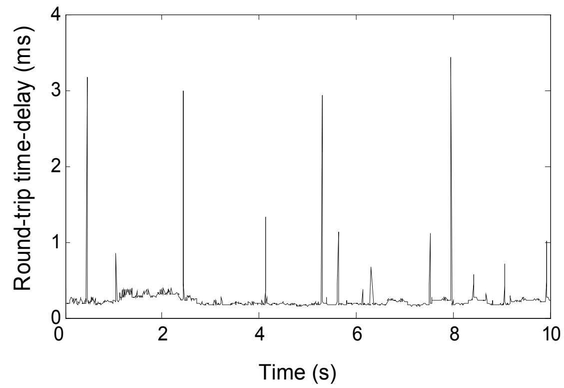

Fig. 3. Round-trip time delays of maglev setup.

A. 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 the data packets 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. 4. 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.

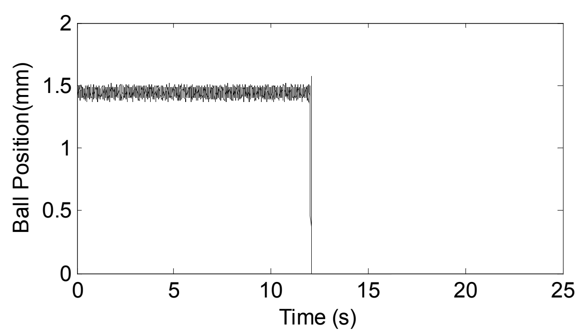

Fig. 4. Ball position with packet losses beginning at t = 12 s.

After conducting many experiments with various models of different orders, the following fifth-order AR model was chosen as the sensor data prediction.

$$\hat{y}(n) = 0.3195y(n-1) + 0.2669y(n-2) - 0.0622y(n-3) + 0.1960y(n-4) + 0.4064y(n-5). \tag{17}$$

The time-delay/packet-loss compensation algorithm (7) was also implemented to compensate for the packet loss from controller node to the actuator node. In the second experiment, artificial packet losses were introduced and 4 successive packet losses occurred every 6 s from t = 3 s onwards. The response of the system is shown in Fig. 5. The NCS test setup maintained its stability successfully with periodic 1.1 mm spikes in the ball position from the estimation error due to the packet losses.

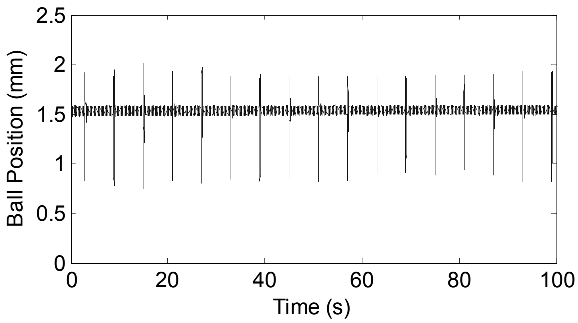

Fig. 5. Ball position with 4 successive packet losses occurring every 6 s.

In the third experiment, the compensation algorithms are implemented. Artificial packet losses at a 20% average packet-loss rate were introduced after t = 12 s. As shown in Fig. 6, the system remained stable throughout the experiment. The performance was degraded due to the multi-step-ahead state estimation that would inevitably introduce estimate errors. This result demonstrates the effectiveness of the developed control strategies.

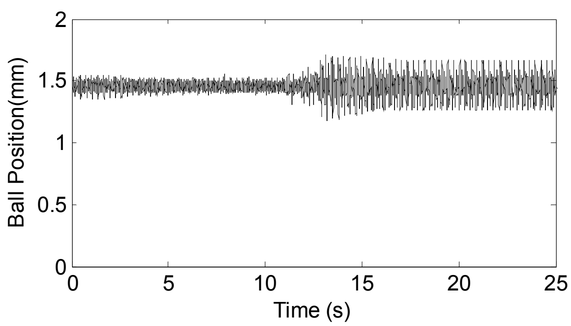

Fig. 6. Ball position with 20% packet losses beginning at t = 12 s onwards.

B. 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. Fig. 7 shows a closed-loop step (started at t = 7 s) response of the test setup without packet loss.

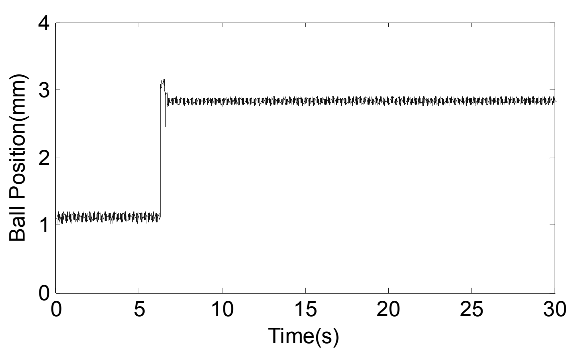

Fig. 7. Step response without packet loss.

Fig. 8 shows the step (started at t = 7 s) response with packet losses. The average packet-loss rate was 20%. The closed-loop system was stable with a worse stability margin due to packet loss and imperfect state estimation.

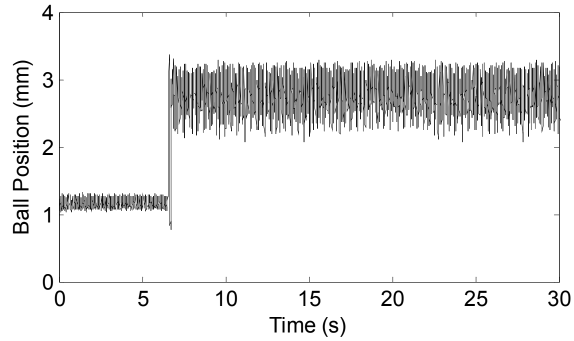

Fig. 8. Step response of system with packet loss occurs from t = 7 s onwards.

Fig. 9 shows the compensated system response of tracking a sinusoidal position command at a 20% packet-loss rate from t = 50 s onwards. Repeating experiments with various frequencies, we conclude that the closed-loop system bandwidth was reduced from 2.7 Hz to 0.34 Hz due to packet losses and imperfect state estimation.

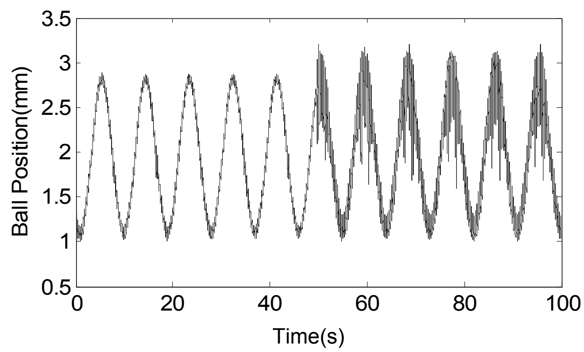

Fig. 9. Tracking a sinusoidal command with 20% packet loss from t = 50 s onwards.

Figs. 10 and 11 show the system response of tracking a sawtooth signal at different packet-loss rates from t = 34 s onwards. Comparing Figs. 10 and 11, we can see that the fluctuation in movement of the ball increases with the increase of the packet-loss rate.

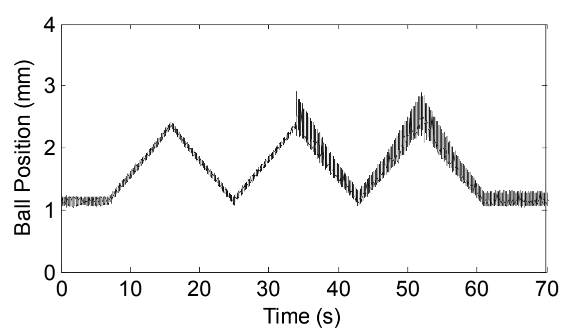

Fig. 10. System tracking response at a 10% packet loss rate.

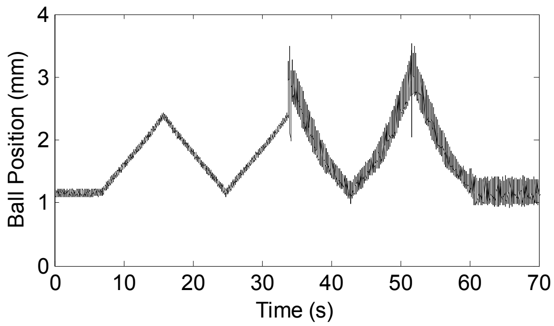

Fig. 11. System tracking response at a 20% packet drop rate.

The above two sets of experimental results demonstrate that the control strategies developed in this paper are effective to maintain the stability and the tracking performance of our open-loop unstable test bed even with 20% packet loss. However, the overall performance is degraded due to packet loss and the multi-step-ahead state estimation that would inevitably introduce estimate errors.

VI. CONCLUSIONS

In this paper, we presented networked real-time control strategies to deal with stochastic time delays and packet losses in NCS. 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 experimental verification. 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 5 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.

REFERENCES

[1] H. Chan and U. Ozguner, "Closed-loop control of systems over a communications network with queues," *International Journal of contorl*, vol. 62, no.3, pp. 493–510, June 1995.

[2] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, Feb. 2003.

[3] G. Schickhuber and O. McCarthy, "Distributed fieldbus and control network systems," *Comput. Contr. Eng.*, vol. 8, no. 1, pp. 21–32, Feb. 1997.

[4] R. S. Raji, "Smart networks for control," *IEEE Spectrum*, vol. 31, no. 6, pp. 49–55, June 1994.

[5] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet and DeviceNet," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66–83, Feb. 2001.

[6] G. C. Walsh, Y. Hong, and L. G. Bushnell, "Stability analysis of networked control system," *IEEE Transactions on Control System Technology*, vol. 10 no. 3, pp. 438–446, May 2002.

[7] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control System gazine*, vol 21, no. 1, pp. 84–99, Feb. 2001.

[8] Y. Halevi and A. Ray, "Integrated communication and control systems: Part I—Analysis," *ASME Journal of Dynamic Systems, Measurement and Control,* vol. 110, no. 4, pp. 367–373, Dec. 1988.

[9] A. Ray and Y. Halevi, "Integrated Communication and Control Systems: Part II—Design Consideration," *ASME Journal of Dynamic System, Measurement and Control*, vol. 110, no. 4, pp. 374–381, Dec. 1988.

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

[11] G. C. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," *Proceedings of the IEEE international conference on control applications*, vol. 2, pp. 1448–1453, Kohala Coast, HI, 1999.

[12] R. Krotolica, 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, Oct. 1994.

[13] L. W. Liou and A. Ray, "A stochastic regulator for integrated communication and control systems: Part I–Formulation of control law," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 113, no. 4, pp. 604–611, Dec. 1991.

[14] J. Nilsson, *Real-Time Control Systems with Delays*, Ph.D dissertation, Dept. Automatic Control. Lund Institute of Technology, Lund, Sweden, Jan. 1998.

[15] F.-L. Lian, J. R. Moyne, and D. M. Tilbury*, 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, July 2001.

[16] N. J. Ploplys and A. G. Alleyne, "UDP network communications for distributed wireless control," *Proc. of 2003 American Control Conference*, pp. 3335–3340, June 2003.

[17] N. B. Almutairi, M.-Y. Chow, and Y. Tipsuwan, "Network-based controlled DC motor with fuzzy compensation," Proc*. Of the 27 th annual conference of the IEEE Industrial Electronics Society*, vol. 3, no. 5, pp. 1844–1849, June 2001.

[18] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: reducing communication in distributed control system using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 503–518, July 2002.

[19] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, the MIT Press, Cambridge, Massachusetts, 1983.

[20] A. Srivastava, *Distributed Real-Time Control via the Internet*, M.S. Thesis, Texas A&M University, May 2003.

[21] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Athena Scientific, Blemont, MA, USA, 1995.

[22] F. Lewis, *Optimal Estimation*, Wiley & Sons, Inc. New York, USA, 1986.

[23] S. C. Paschall, II, *Design, Fabrication and Control of a Single Actuator Magnetic Levitation System*, Senior Honors Thesis, Texas A&M University, May 2002.

[24] http://www.rtai.org.