European Journal of Control (2007)4:1–8

European Journal of Control

Stabilization of Networked Control System with Time Delays and Data-Packet Losses

Kun Ji and Won-jong Kim*

Department of Mechanical Engineering, Texas A&M University, College Station, Texas, USA

In this paper, we provide experimental results in the application of the results of [4]. Based on the theoretic work in [4], we propose an estimator-based delay-compensation algorithm to stabilize a networked control system (NCS) with network-induced stochastic time delays, data-packet losses, and out-of-order data-packet transmissions. With the p-sampling-period delay upper bound, the NCS can also accommodate up to $p-1$ successive packet losses. We also derive sufficient conditions for the stability of the NCS. The feasibility and effectiveness of the theoretic results of [4] are verified experimentally using an NCS test bed incorporating an open-loop unstable ball magnetic-levitation (maglev) system we constructed.

Keywords: Data-packet losses; model-based estimation; networked control system; time delays

1. Introduction

A control system where sensors, actuators, and controllers are interconnected over a communication network is called an networked control system (NCS). Randomly-varying time delays induced by the network are well known to degrade the system stability and performance [6]. The effect of time delays or data-packet losses on the stability and performance of control systems has been a subject of many studies. Delchamps [2] investigated the issue of stabilizing a discrete-time linear system with quantized state feedback. The problem of state estimation and stabilization of a linear-time-invariant (LTI) system with a finite-bandwidth digital communication channel capacity was introduced by Wong and Brockett [9]. In [8], the stability of NCSs for a continuous-time plant and controller was studied. The network resides only between the plant and the controller. A study on stability and performance of feedback control systems with multiple time delays was reported in [1] considering the roots of the closed-loop characteristic equation. In [3], a robust state-predictive control strategy was proposed for discrete-time multi-input/output systems with non-equal delays on signal buses. The input and output delays were taken into account in the control law synthesis and a steady-state Kalman predictor design. Zhang et al. [10] assumed that the full states were transmitted through the network and that they might be lost because of the dropped packets in the network. On the other hand, the control command was sent directly to the plant. With these assumptions, the authors used the stability analysis for asynchronous dynamical systems to find the maximum packet-dropping rate under which the overall system is stable. In [7], the stability of a linear NCS in the presence of dropped packets was studied. Similar to [10], the controller was directly connected to the plant, so there was direct link between the controller and the plant. The stability analysis in [7] was based on the stability of Markov-jump linear systems. Almost all the above

*Correspondence to: Won-jong Kim, E-mail: wjkim@tamu.edu

Received 28 December 2004; Accepted 8 May 2007
Recommended by V. Rasvan, A. Isidori

research focused on dealing with time delays or packet losses separately. Network-induced time delays and packet losses may occur simultaneously, thus a more comprehensive compensation algorithm is needed that can deal with the time delays and data-packet losses in a unified way.

In [4], the control of a continuous-time linear plant where the state sensor was connected to a linear controller/actuator via a network was addressed, and necessary and sufficient conditions for stability were derived for the presented setup in terms of the update time h and the parameters of the plant. In this paper, we experimentally verify the results in [4] using our NCS test bed. We consider an NCS framework as shown in Fig. 1. Two classes of time delays are included 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. We do not presume that this network is exclusively dedicated to the NCS, so other types of data traffic are allowed to use the same communication medium.

Based on the model of [4], we propose herein a model-based estimation algorithm to compensate for network-induced time delays and data-packet losses in both the forward path and the feedback path. A model estimator [4] is used at the controller node, and it re-creates the plant dynamics for state estimation in the forward path. The states are estimated multiple steps in advance with the model plant dynamics. Control data are also estimated multiple steps in advance based on these estimated states and sent to the actuator node in one packet. The actuator node selects a single appropriate control signal from this packet in a given sampling time interval. By this way and using an event-driven controller and estimator, the plant receives a control input at each sampling time interval, and network-induced data-packet losses have no more adverse effect than network-induced time delays. They both can be compensated for with the same algorithm in a unified way.

In Section 2 we present the problem statement. In Section 3 a control algorithm for time-delay and packet-loss compensation is presented. Experimental results that verify the stabilization algorithm and the stability conditions in [4] are presented in Section 4.

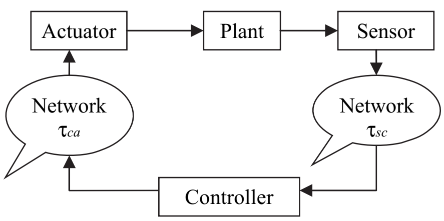

Fig. 1. An NCS framework with network-induced round-trip time delays.

2. Problem Statement

In this paper, the following assumptions are made:

1. The sensor node and the actuator node are time-driven, and the controller node is event-driven.
2. The upper bound of the total time delay is less than p sampling intervals, and the number of consecutive packet losses is less than $p-1$ in both the feedback path and the forward path.

2.1. Full-State-Feedback NCS Case

If all the states are measurable, the sensor mode can send the full states through the network to update the model estimator as shown in Fig. 2. Assume that the sampling period is T, then we have following difference equations:

Discrete-time plant model:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n) \end{aligned} \tag{1}$$

Model estimator:

$$\begin{aligned} \hat{x}(n+1) &= \hat{A}\hat{x}(n) + \hat{B}u(n) \\ \hat{y}(n) &= \hat{C}\hat{x}(n) + \hat{D}u(n) \end{aligned} \tag{2}$$

Full-state-feedback controller:

$$u(n) = K\hat{x}(n) \tag{3}$$

State-estimation error:

$$e(n) = x(n) - \hat{x}(n), \qquad n \in [n_k, n_{k+1}). \tag{4}$$

We define a positive integer variable, $N(k) = n_{k+1} - n_k$, where k is the index of events, n is the index of sampling instants, and n_k is the index of event arrival

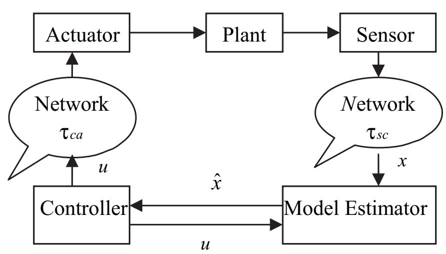

Fig. 2. Block diagram of a full-state-feedback NCS.

instants. The "event" herein means that the sensor data arrived at the controller node. Thus,

$$e(n_k) = x(n_k) - \hat{x}(n_k) = 0. \tag{5}$$

In other words, the state error in the model estimator is reset to 0 at the n_k-th instant when the actual sensor data are available.

The $N(k)$ has the following components

$$N(k) = N_{\text{tran}} + N(k)_{\text{delay}}, \tag{6}$$

where N_{tran} is the time required for the data transmission from the sensor node to the model estimator, and $N(k)_{\text{delay}}$ is the time for other total delays, such as preparing time, waiting time, and processing time at the network nodes. The upper bound of $N(k)$ is p by Assumption 2.

2.2. Output-Feedback NCS Case

When it is impossible to directly measure all the plant states, we may implement a state observer as shown in Fig. 3 [4]. This observer sends the observed states to the model estimator. Similarly, with the same dynamic Eqs (1–3) for the observer,

$$\tilde{x}(n+1) = (\hat{A} - L\hat{C})\tilde{x}(n) + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} u(n) \\ y(n) \end{bmatrix}, \tag{7}$$

where $\tilde{x}(n)$ is the observer states, and L is the observer gain. Define $\bar{e}(n) = \tilde{x}(n) - \hat{x}(n)$, we have

$$\bar{e}(n) = \begin{cases} \tilde{x}(n) - \hat{x}(n), & n \in (n_k, n_{k+1}) \\ 0, & n = n_k \end{cases}. \tag{8}$$

The above models are similar to the NCS models in [4], in order to deal with the case where there are time delays and data-packet losses in both the feedback path and the forward path as shown in Fig. 1, we develop the algorithms to compensate for these two classes of time delays and data-packet losses simultaneously and then the stability conditions proposed in [4] can be applied to the NCS framework shown in Fig. 1.

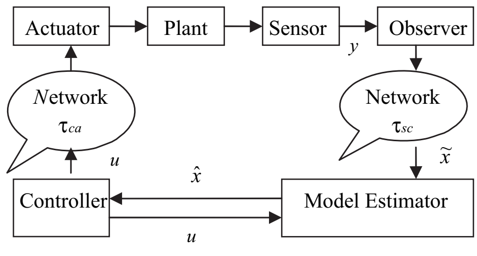

Fig. 3. Block diagram of an output-feedback NCS.

3. Compensation Algorithm for Network-Induced Time Delays and Data-Packet Losses

The delay τ_{sc} can be compensated for if we consider the controller and the estimator as event-driven devices. The estimator receives an updated state $x(n_k)$ after $N(k)$ sampling intervals. The remaining problem is how to compensate for the effect of delay τ_{ca} or packet losses in the forward path to ensure the plant receives a proper control data in every sampling interval.

3.1. Compensation Algorithm for τ_{ca} and Data-Packet-Loss Compensation

We use an estimator as proposed in [4] at the controller node to estimate the plant states of the successive samples p-step in advance. With these estimated states the controller node calculates the control signals p-steps ahead and then sends them in one packet to the actuator node. If the new control signal packet does not arrive in time, the actuator node uses estimated control signals from the control signal packet that previously arrived.

The p-step-ahead state estimation is done as

$$\begin{aligned} \hat{x}(n+1) &= \hat{A}\hat{x}(n) + \hat{B}u(n) \\ \hat{x}(n+2) &= \hat{A}\hat{x}(n+1) + \hat{B}\hat{u}(n+1) \\ &\dots \\ \hat{x}(n+p) &= \hat{A}\hat{x}(n+p-1) + \hat{B}\hat{u}(n+p-1). \end{aligned} \tag{9}$$

The control signal packet is generated as

$$\begin{aligned} u(n) &= K\hat{x}(n) \\ \hat{u}(n+1) &= K\hat{x}(n+1) \\ &\dots \\ \hat{u}(n+p-1) &= K\hat{x}(n+p-1). \end{aligned} \tag{10}$$

The actuator node selects an appropriate single control signal $U(k)$ from the packet as below for the next p-sampling intervals until the new updated control signal packet arrives.

$$U(k) = \begin{cases} u(n), & nT \le t < (n+1)T \\ \hat{u}(n+1), & (n+1)T \le t < (n+2)T \\ \dots & \\ \hat{u}(n+p-1), & (n+p-1)T \le t < (n+p)T, \end{cases} \tag{11}$$

where $U(k)$ denotes the actual control signal adopted by the actuator; $u(n)$, $\hat{u}(n+1)$, …, and $\hat{u}(n+p-1)$, the components of the control signal packet at the corresponding sampling time instant n; and t, the continuous time. Once the new control signal packet arrives, Eq. (11) it is revised with the new control signal components, and $U(k+1)$ is then available. An actual real-time implementation of this control law is

$$
\begin{aligned}
U(k) = {} & (1/4)(1+\mathrm{sgn}(t-nT))(1-\mathrm{sgn}(t-(n+1)T))u(n) \\
& + (1/4)(1+\mathrm{sgn}(t-(n+1)T))(1-\mathrm{sgn}(t-(n+2)T))\hat{u}(n+1) \\
& + (1/4)(1+\mathrm{sgn}(t-(n+2)T))(1-\mathrm{sgn}(t-(n+3)T))\hat{u}(n+2) \\
& + \dots \\
& + (1/4)(1+\mathrm{sgn}(t-(n+p-1)T))(1-\mathrm{sgn}(t-(n+p)T))\hat{u}(n+p-1),
\end{aligned}
\tag{12}
$$

where $\mathrm{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$. The compensation for time delays and packet losses with this algorithm is elaborated in the following.

For example, with $p=4$, the adoptions of the control signals by the actuator node in each sampling interval when packet losses occur are illustrated in Fig. 4. The label y denotes the sensor data, u, the control data, and T, the sampling period. The labels $U*$ denote the control law Eq. (12) implementation at the actuator node. The subscripts of the labels y, u, and T denote the indices of sampling intervals as n in Eq. (1). The subscripts of the label U denote the indices of events as k in Eq. (5). The subscript of the label u also indicates whether it is estimated or not. If no new control signal packet is available in any given sampling interval, the most recently estimated control signal and stored in the last available control signal packet, like $\boldsymbol{u_{22e}}$ and $\boldsymbol{u_{3e}}$ (in bold face) in Fig. 4 is used.

The adoptions of the control signals by the actuator node in each sampling interval when non-consecutive time delays occur are illustrated in Fig. 5. For non-consecutive time delays, the delayed packet likes U_2 in Fig. 5 is simply discarded. The most recently arrived control packet is used.

On the other hand, for consecutive time delays, the delayed packet is still useful. As shown in Fig. 6, the delayed and used control signal $\boldsymbol{u_{3e}}$ in the second packet is more recent than the one in the first packet. The control signal $\boldsymbol{u_{4e}}$ in the third packet is more recent than the one in the first packet or the second packet. Thus the most recently estimated control signal is used.

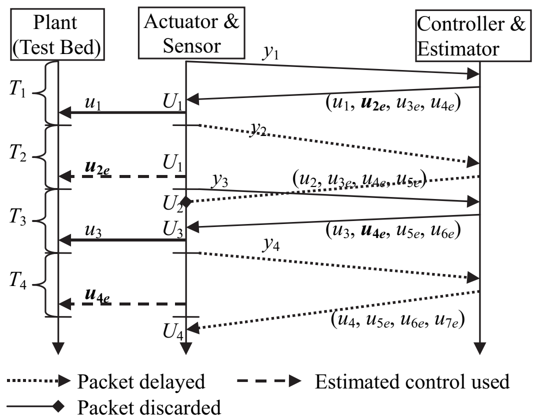

Fig. 5. Example communication process with non-consecutive time delays.

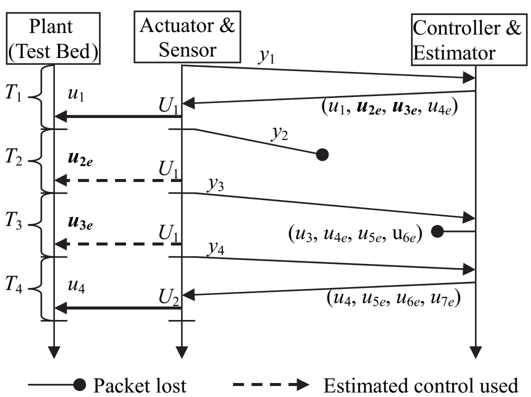

Fig. 4. Example communication process with packet losses.

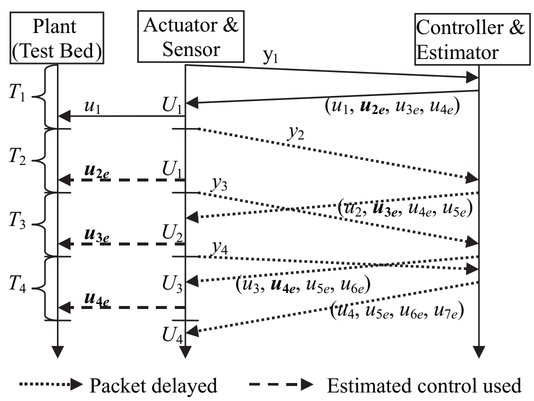

Fig. 6. Example communication process with three consecutive time delays.

In the case of out-of-order transmission of packets, the outdated packets are simply discarded. For instance, if the $(n+1)$-th control signal packet arrives earlier than the n-th packet at the actuator node from the controller node, the actuator node neglects the n-th packet and uses the most recent $(n+1)$-th control signal packet. By this, our algorithm can deal with out-of-order packet transmission as well. An example of such communication process is shown in Fig. 7 where the outdated sensor data y_2 and the control packet U_3 are discarded.

Fig. 8 shows an example communication process that includes all the above situations. The labels in bold face such as $\boldsymbol{u_{2e}}$, $\boldsymbol{u_{3e}}$, and so on denote the estimated control signals. The labels in unbold face such as u_1, u_5, and u_7 denote the real control signals. In Fig. 8, the numerical indices of the control signals the plant receives is exactly the same as those of the sampling intervals, thus we can see that in every sampling interval, the plant receives a proper control signal that is either real or most recently estimated in the previous sampling intervals. By this algorithm, the effect of packet losses is no worse than that of time delays. Thus this algorithm compensates for time delays, packet losses, and out-of-order packet transmissions in a unified way.

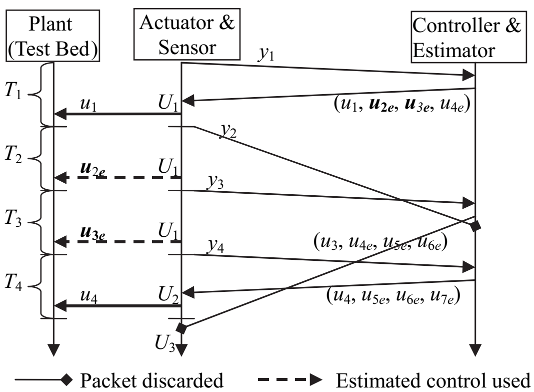

Fig. 7. Example communication process with out-of-order packets arrivals.

3.2. Augmented System Equations

With the compensation algorithm developed above, the plant can receive the control input $u(n) = K\hat{x}(n)$ at each time instant n. Thus we can develop following augmented system equation to facilitate the stability-conditions developed in [4].

Let $\bar{A} = A - \hat{A}$, $\bar{B} = B - \hat{B}$, $\bar{C} = C - \hat{C}$, and $\bar{D} = D - \hat{D}$ be the model error matrices.

1. For the full-state-feedback NCS shown in Fig. 2, the augmented system equation is

$$\begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A+BK & -BK \\ \bar{A}+\bar{B}K & \hat{A}-\bar{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}. \quad (13)$$

Define $z_1(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}$, $\Lambda_1 = \begin{bmatrix} A+BK & -BK \\ \bar{A}+\bar{B}K & \hat{A}-\bar{B}K \end{bmatrix}$, then (13) can be rewritten as

$$z_1(n+1) = \Lambda_1 z_1(n), \quad n \in [n_k, n_{k+1}). \quad (14)$$

The system described by Eq. (14) with the initial conditions $z_0 = [\,x(n_0) \quad 0\,]^T$ has the following solution [4].

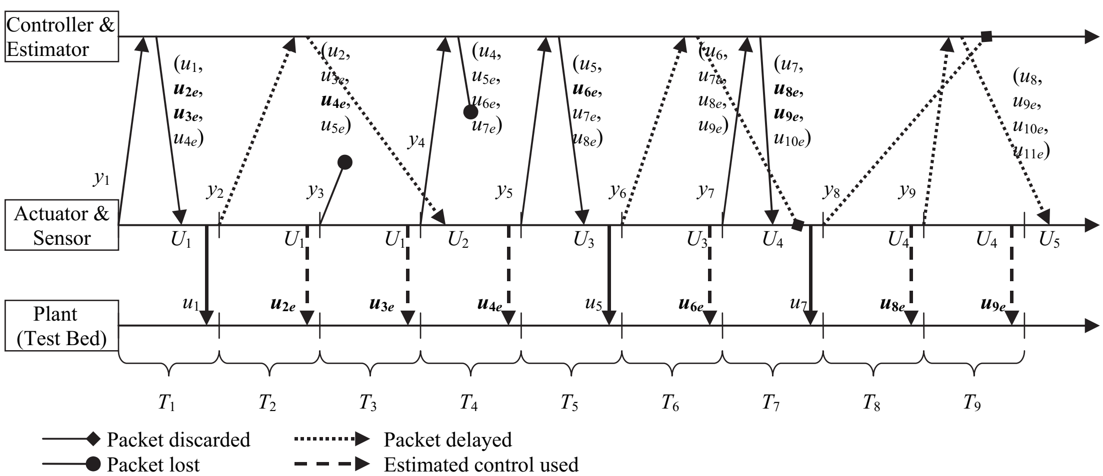

Fig. 8. Example communication process with time delays, packet losses, and out-of-order packet arrivals altogether.

$$z_1(n) = \Lambda_1^{n-n_k} \prod_{i=0}^{k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_1^{N(i)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right) z_0 \quad (15)$$

2. For the output-feedback NCS shown in Fig. 3, the augmented system equation is

$$\begin{bmatrix} x(n+1) \\ \tilde{x}(n+1) \\ \bar{e}(n+1) \end{bmatrix} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\bar{D}K & -\hat{B}K - L\bar{D}K \\ LC & L\bar{D}K - L\hat{C} & \hat{A} - L\bar{D}K \end{bmatrix} \begin{bmatrix} x(n) \\ \tilde{x}(n) \\ \bar{e}(n) \end{bmatrix}. \quad (16)$$

Define

$$\Lambda_2 = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\bar{D}K & -\hat{B}K - L\bar{D}K \\ LC & L\bar{D}K - L\hat{C} & \hat{A} - L\bar{D}K \end{bmatrix},$$

and $z_2(n) = \begin{bmatrix} x(n) \\ \tilde{x}(n) \\ \bar{e}(n) \end{bmatrix}$, then we have

$$z_2(n+1) = \Lambda_2 z_2(n), \quad n \in [n_k, n_{k+1}). \quad (17)$$

The system described by Eq. (17) with the initial condition $z_0 = [\, x(n_0) \quad \tilde{x}(n_0) \quad 0\,]^T$ has the following solution [4].

$$z_2(n) = \Lambda_2^{n-n_k} \prod_{i=0}^{k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda_2^{N(i)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) z_0 \quad (18)$$

The sufficient conditions for the stability of full-state-feedback and output-feedback NCSs proposed in [4] can be used here as sufficient conditions for Eqs (15) and (18).

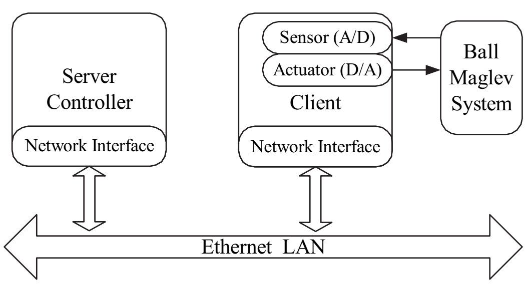

Fig. 9. Block diagram of our NCS test bed.

4. Experimental Results

To verify the effectiveness of the estimation algorithm developed in [4] and the compensation method mentioned in Section 3, we set up an NCS test bed of a ball maglev system as shown in Fig. 9 [11]. The control objective is to levitate a steel ball at a predetermined steady-state equilibrium position with an electromagnet. It is an open-loop unstable system, and the system stability will be lost if any control actuation misses its deadline due to delayed or lost control data packets. Thus this test bed is very suitable for the verification purpose.

For our ball maglev system, it is found the upper bound of the network induced round trip time delay is about five sampling periods. Thus we chose to design a 4-step-ahead estimator, and the composition of a 56-byte-long Internet-protocol (IP) packet transmitted from the server to the client as shown in Fig. 10. It consists of a 20-byte-long IP header, an 8-byte-long user datagram protocol (UDP) header, an 8-byte-long time stamp, one current control signal, and four estimated control signals.

In the first experiment, no compensation algorithm was used. At $t = 10$ s, we forced a data packet to be lost while transferred from the client to the server. With the introduction of this data-packet loss, it was expected that the system would become unstable. In the response of the system shown in Fig. 11, the 0 value after $t = 10$ s represents that the system indeed lost its stability. The ball could not maintain its equilibrium position and fell down.

In the second experiment, the estimator-based compensation algorithm was implemented, and one packet loss was introduced after every four successful data transmissions (i.e, at s 20% packet-loss rate) from $t = 10$ s onwards. As evident from the response shown in Fig. 12, the system remained stable with degraded performance, but the ball did not fall down from its equilibrium position. In Fig. 12, we can observe some hidden oscillations. In practical NCS applications, asynchronous and aperiodic sampling often occurs since the computer is time-shared or a part of a computer network, or there are technical imperfections in the instrumentation. However, asynchronous and

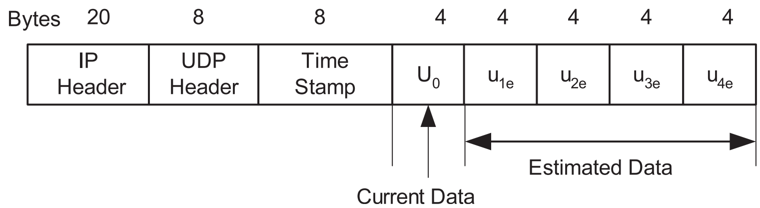

Fig. 10. Composition of a control-data packet from the controller to the actuator.

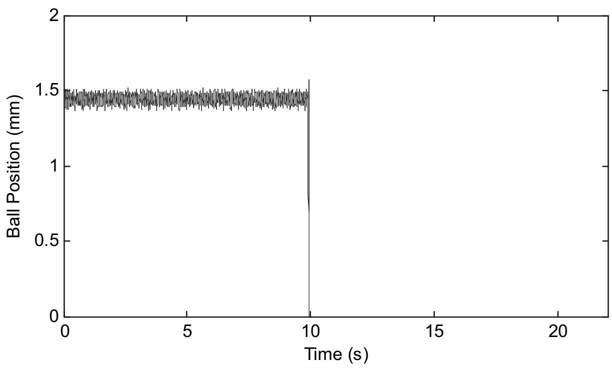

Fig. 11. Ball position with a packet loss occurring at $t = 10\,\mathrm{s}$ without the compensation algorithm.

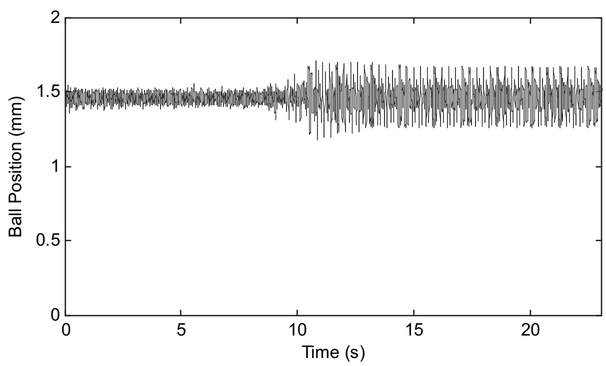

Fig. 12. Ball position with 20% packet losses occurring from $t = 10\,\mathrm{s}$ onwards with the compensation algorithm.

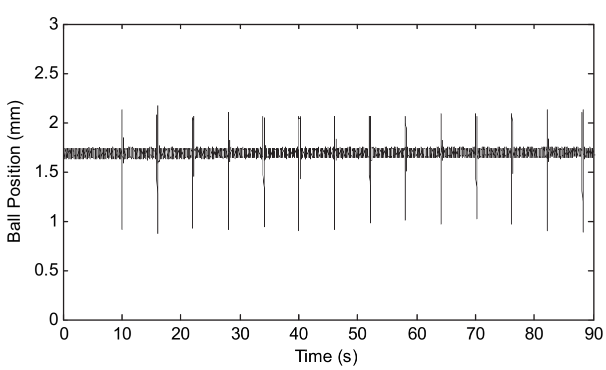

Fig. 13. Ball position with four successive packet losses occurring every 6 s from $t = 10\,\mathrm{s}$ onwards with the compensation algorithm.

aperiodic sampling is sometimes preferred over synchronous sampling when the former is applied intentionally to eliminate hidden oscillations.

In the third experiment, the compensation algorithm was implemented and four consecutive packet losses were introduced every 6 s from $t = 10\,\mathrm{s}$ onwards. The response of the system is shown in Fig. 13. The system maintained its stability successfully even in the event of four successive packet losses, and the ball did not fall down from its equilibrium position. The periodic 1.1-mm peak-to-peak spikes indicate the degraded system performance due to the packet losses and imperfect state estimation.

The above experimental results demonstrated that the system stability in the presence of time delays or packet losses could be maintained using the estimator-based compensation algorithm developed in Section 3.

5. Conclusions

In this paper, based on the model in [4], we proposed an estimator-based delay-compensation algorithm to stabilize an NCS in the presence of network-induced time delays and data-packet losses. This model-based estimation algorithm can deal with network-induced stochastic time delays, data-packet losses, and out-of-order packet transmissions in a unified way.

We incorporated a controllable and observable maglev test bed that is open-loop unstable in our NCS test setup. We provided experimental results in the application of the results of [4], and the experimental results demonstrated the feasibility and effectiveness of this estimator-based compensation algorithm for NCSs with stochastic time delays and successive packet losses. We could stabilize our open-loop-unstable ball maglev system even with time delays up to five sampling intervals and four consecutive packet losses, or at an average packet-loss rate up to 20%.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant

No. CMS-0116642. The authors thank A. Ambike and S.C. Paschall for their contribution in this research.

References

1. Ali MS, Hou ZK, Noori MN. Stability and performance of feedback control systems with time delays. Comput and Struct 1998; 66: 241–248
2. Delchamps DF. Stabilizing a linear system with quantized state feedback. IEEE Trans Autom Control 1990; 35: 916–924
3. Marinescu B, Bourlès, H. Robust state-predictive control with separation property: a reduced-state design for control systems with non-equal time delays. Autom 2000; 36: 555–562
4. Montestruque LA, Antsaklis PJ. Model-based networked control systems stability. ISIS Technical Report ISIS-2002–001, 2001
5. Paschall SC. Design, fabrication and control of a single actuator magnetic levitation system. Senior Honors Thesis, Texas A&M University, College Station, TX, 2002
6. Ray A, Halevi Y. Integrated communication and control systems: part I–analysis. ASME J Dyn Syst Meas Control 1988; 110: 367–373
7. Seiler P, Sengupta R. Analysis of communication losses in vehicle control problems. In Proceedings of the American Control Conference, Arlington, VA, June 25–27, 2001; 1491–1496
8. Walsh GC, Hong Y, Bushnell LG. Stability analysis of networked control systems. IEEE Trans Control Syst Technol 2002; 10: 438–446
9. Wong WS, Brochett RW. Systems with finite communication bandwidth–Part II: stabilization with limited information feedback. IEEE Trans Autom Control 1999; 44: 1049–1053
10. Zhang W, Branicky MS, Phillips SM. Stability of networked control systems. IEEE Control Syst Mag 2001; 21: 84–99