17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Performance of Model-Based Networked Control Systems with Discrete-Time Plants

Tomas Estrada Panos J. Antsaklis*

Abstract

In this paper we study performance-related aspects for plants in a networked control setting, employing an approach known as Model-Based Networked Control Systems (MB-NCS) with Intermittent Feedback. Model-Based Networked Control Systems use an explicit model of the plant in order to reduce the network traffic while attempting to prevent excessive performance degradation. Intermittent Feedback consists of the loop remaining closed for some time interval, then open for another interval. We begin by investigating the behavior of the system while tracking a reference input. We provide the full response of the system and a condition for stability. We then shift our attention to controller design for MB-NCS. We use dynamic programming techniques to design an optimal controller to optimize an LQ-like performance index.

1. Introduction

A networked control system (NCS) is a control system in which a data network is used as feedback media. NCS is an important area in control, see for example recent surveys such as [2] and [11], as well as [23], [27], and [28]. The use of networks as media to interconnect the different components of an industrial system is rapidly increasing. However, the use of NCSs poses some challenges. One of the main problems to be addressed when considering an NCS is the size of the bandwidth required by each subsystem. A particular class of NCSs is model-based networked control systems (MB-NCS), introduced in [17]. The MB-NCS architecture makes explicit use of the knowledge of the plant dynamics to enhance the performance of the system. Here we extend this work by taking advantage of the concept of intermittent feedback. In the previous work done in MB-NCS, the updates given to the model of the plant state were performed in instantaneous fashion, but with intermittent feedback the system remains in closed loop control mode for more extended intervals. This notion makes sense as it is a good representation of what occurs in both nature and industry. For example, when driving a car, when approaching a curve or hilly terrain, we pay attention to the road for a longer time, which is equivalent to staying in closed-loop mode, and we only reduce our attention -switch to open loop control- when the road is once again straight. It is worth noting that while the application of intermittent feedback to MB-NCS is novel, the concept has been studied in a variety of fields such as [13], [23],[26], [14], [21]. While intermittent control is a very intuitive notion, its combination with the MB-NCS architecture allows for obtaining important results and opening new paths in controlling NCSs effectively.

In previous work [6, 7, 8], we have provided stability results for both the continuous and discrete setting. In this paper we shift our attention towards performance related aspects. We begin by considering an MB-NCS architecture with a reference input. In this architecture, the model receives its input directly from the controller, which is different from the plant's input. We provide a full description of the state response of the system and a condition for stability. The results are derived for instantaneous feedback, but by taking $\tau = 0$ they apply for the instantaneous feedback case as well. We then consider a discrete time MB-NCS without reference input and design an optimal controller to meet a linear quadratic-like performance index. Our methodology is based on the dynamic programming approach and is similar to that presented in [24] and [25].

The rest of the paper is organized as follows. In Section 2, we investigate the behavior of MB-NCS model-based for the first architecture. We present the problem formulation in detail, derive the complete description of the state response and present a stability condition. In Section 3, we provide a method to design an optimal controller for an MB-NCS, by using dynamic programming techniques. Finally, in Section 4, we provide conclusions and propose future work.

*T. Estrada and P.J. Antsaklis are with the the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: {testrada,antsaklis.1}@nd.edu

2. Discrete-time MB-NCS with Intermittent feedback and Reference Input

We will presently study the behavior of an MB-NCS with instantaneous feedback, introducing a reference input signal r. We will consider two cases, differing from each other in the implementation of the input to the model. The first one is depicted in the following figure.

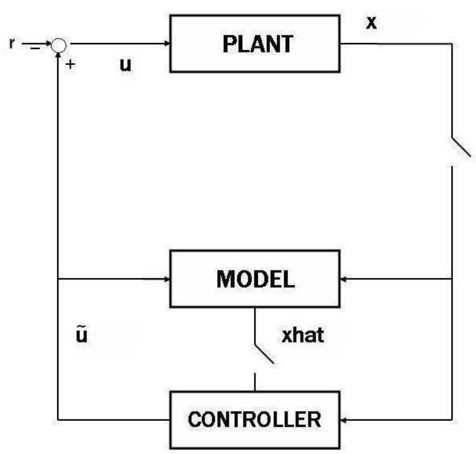

Figure 1. Basic MB-NCS architecture

In this architecture, the model receives its input directly from the controller, which is different from the plant's input. This corresponds to a situation where the model is collocated with the controller and set in a remote access location.

2.1. State response of the system

The plant is given by $x(n+1) = Ax(n) + Bu(n)$, the plant model by $\hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}\hat{u}(n)$. The controller is linear state feedback, with the difference that while the input to the controller is still $\hat{u}(n) = K\hat{x}(n)$, the input to the plant is now given by $u(n) = K\hat{x}(n) - r(n)$.

The state error is defined as $e = x - \hat{x}$ and represents the difference between plant state and the model state. The modeling error matrices $\tilde{A} = A - \hat{A}$ and $\tilde{B} = B - \hat{B}$ represent the plant and the model. We also define the vector $z(n) = [x^T \; e^T]^T$.

We will now proceed to derive the response and later summarize the result in a proposition.

As in the case without reference input studied in [6], the error is reset every h seconds, the loop remains closed for τ seconds, and the system runs open loop for a period of $\tau - h$ seconds. Recall that in the discrete time case, τ and h are both integers.

Let us first consider what happens during the open loop interval, that is, when $n \in [n_k + \tau, n_{k+1})$. For this interval, we have that

$$u = K\hat{x} - r \tag{1}$$

so

$$\begin{bmatrix} x(n+1) \\ \hat{x}(n+1) \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ \hat{x}(n) \end{bmatrix} + \begin{bmatrix} -B \\ -\hat{B} \end{bmatrix} r \tag{2}$$

with initial conditions $\hat{x}(n_k + \tau) = x(n_k + \tau)$.

Rewriting in terms of x and e, that is, of the vector z:

$$z(n) = \begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} + \begin{bmatrix} -B \\ -B \end{bmatrix} r \tag{3}$$

$$z(n_k) = \begin{bmatrix} x(n_k) \\ e(n_k) \end{bmatrix} = \begin{bmatrix} x(n_k^-) \\ 0 \end{bmatrix}, \forall n \in [n_k + \tau, n_{k+1})$$

Thus, we have

$$z(n+1) = \Lambda_o z(n) + \Psi r(n), \tag{4}$$

$$\text{where } \Lambda_o = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix}$$

$$\text{and } \Psi = \begin{bmatrix} -B \\ -B \end{bmatrix}, \forall n \in [n_k + \tau, n_{k+1})$$

For the closed loop case, similarly we obtain

$$z(n+1) = \Lambda_c z(n) + \Psi r(n), \tag{5}$$

$$\text{where } \Lambda_c = \begin{bmatrix} A + BK & -BK \\ 0 & 0 \end{bmatrix}$$

$$\text{and } \Psi = \begin{bmatrix} -B \\ -B \end{bmatrix}, \forall n \in [n_k, n_k + \tau)$$

From this, it should be quite clear (by solving the difference equation) that given an initial condition $z(n = 0) = z_0$, then after a certain time $n \in [0, \tau)$, during which the system has been running in close loop, the solution of the trajectory of the vector is given by

$$z(n) = \Lambda_c^n z_0 + \sum_{j=0}^{n-1} \Lambda_c^n z_{j+1} \Psi r(j), \; n \in [0, \tau). \tag{6}$$

Notice that this evidently can be expressed as the sum of two terms, a "zero-input response" and a "zero-state response".

Similarly, for the first open loop interval, the solution of the trajectory is given by

$$z(n)=\Lambda_o^{n-\tau}z(\tau)+\sum_{j=\tau}^{n-1}\Lambda_o^{n-\tau}z_{j+1}\Psi r(j)\,,\ n\in[\tau,h)\quad(7)$$

Notice that this, too, can be expressed as the sum of a "zero-input response" and a "zero-state response". We will thus continue the derivation by focusing first on what happens to the zero-input response, then to the zero-state response.

The zero-input response is identical to that which we had in the case without reference input, described in [8].

After k cycles, going through this analysis yields a solution

$$z(n_k)=\left(\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\Lambda_o(h-\tau)\Lambda_c(\tau)\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\right)^k z_0 = \Sigma^k z_0\,,$$

where $\Sigma=\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\Lambda_o(h-\tau)\Lambda_c(\tau)\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}$.

The final step is to consider the last (partial) cycle that the system goes through, that is, the time $n\in[n_k,n_{k+1})$. If the system is in closed loop, that is, $n\in[n_k,n_k+\tau)$, then the solution can be achieved merely by pre-multiplying $z(n_k)$ by $\Lambda_c(n-n_k)$. In the case of the system being in open loop, that is, $n\in[n_k+\tau,n_{k+1})$, then clearly we must pre-multiply by $\Lambda_o(n-(n_k+\tau))\Lambda_c(\tau)$.

Now, the zero-state term will experience a similar evolution.

At time τ, the value of the zero-state part will be $\sum_{j=0}^{\tau-1}\Lambda_c^{\tau}z_{j+1}\Psi r(j)$.

Unlike the case with the zero-input case, the portion from the next time interval, rather than premultiplied, will rather be added to this term. So at time h, the value of the zero-state part wil be $\sum_{j=\tau}^{h-1}\Lambda_o^{h-\tau}z_{j+1}\Psi r(j)+\sum_{j=0}^{\tau-1}\Lambda_c^{\tau}z_{j+1}\Psi r(j)$.

Notice that the error portion of the term will have to be reset at each n_k, which corresponds to premultiplying by $\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}$.

We continue the add the portion from the next time interval and do so again for k cycles to obtain:

$$z(n)=\sum_{j=0}^{k-1}\Lambda_c^{n-n_k}z_{j+1}\Psi r(j) + \sum_{l=0}^{k-1}\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\begin{pmatrix} \sum_{j=n_l+\tau}^{n_l+h-1}\Lambda_o^{h-\tau}z_{j+1}\Psi r(j)\\ +\sum_{j=n_l}^{n_l+\tau-1}\Lambda_c^{\tau}z_{j+1}\Psi r(j)\end{pmatrix},$$

for $n\in[n_k,n_k+\tau)$, with $n_{k+1}-n_k=h,\ k=1,2,3...$

and similarly for for $n\in[n_k,n_k+\tau)$.

We can combine what we have developed for zero-state and zero-input and summarize the results in the following proposition.

Proposition 1 *The system described by (4) with initial conditions* $z(t_0)=\begin{bmatrix} x(t_0)\\ 0\end{bmatrix}=z_0$ *has the following response:*

$$\begin{aligned} z(n) =&\\ &\Lambda_c(n-n_k)\left(\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\Lambda_o(h-\tau)\Lambda_c(\tau)\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\right)^k z_0\\ &+\sum_{j=0}^{k-1}\Lambda_c^{n-n_k}z_{j+1}\Psi r(j) \qquad (8)\\ &+\sum_{l=0}^{k-1}\left(\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\begin{pmatrix} \sum_{j=n_l+\tau}^{n_l+h-1}\Lambda_o^{h-\tau}z_{j+1}\Psi r(j)\\ +\sum_{j=n_l}^{n_l+\tau-1}\Lambda_c^{\tau}z_{j+1}\Psi r(j)\end{pmatrix}\right)\end{aligned}$$

for $n\in[n_k,n_k+\tau)$,*with* $n_{k+1}-n_k=h,\ k=1,2,3...$, *where* $\Lambda_o=\begin{bmatrix} A+BK & -BK\\ \tilde{A}+\tilde{B}K & \hat{A}-\tilde{B}K\end{bmatrix}$, *and* $\Psi=\begin{bmatrix} -B\\ -B\end{bmatrix}$.

The result is similar for $n\in[n_k+\tau,n_k+1)$ *and is omitted for reasons of space.*

2.2. BIBO stability condition

Having written the state response of the system, we can easily see that we can arrive at a BIBO stability condition. This condition would be the same as the condition for global exponential stability in the case without reference input. This can be clearly seen from the fact that by bounding the input, the zero-state part of the solution will also be bounded, so we need only that the stability of the zero-input part be stable, and this has been done in previous work. We include the theorem here for completeness.

Theorem 2 *The system described by above is BIBO stable around the solution* $z=\begin{bmatrix} x\\ e\end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\Sigma\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}$ *are strictly inside the unit circle, where* $\Sigma=\Lambda_o(h-\tau)\Lambda_c(\tau)$.

The proof can be found in [8]. The results can be extended to other reference input architectures, as in [9]. Also, while we have considered intermittent feedback, the results also apply to the instantaneous feedback case by specifying $\tau=0$. In the next section, we will focus on the instantaneous feedback case to design a controller for a discrete time MB-NCS.

3. Controller design of MB-NCS using optimal control techniques

Let us now consider a system with a discrete-time linear time-invariant plant, where the full information of the state of the plant is only available with probability $\bar{\nu}$. We will consider the case without a reference input in this section. When the state of the plant is not available, the model is used to compute the control action. Our main objective is to find a controller that will optimize an LQ-based performance criterion for this system and to compute this optimal cost as well.

The setup is as follows. We will consider the discrete-time plant governed by:

$$x_{k+1} = Ax_k + Bu_k \tag{9}$$

The equations of the model are given by:

$$\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{B}u_k \tag{10}$$

and the control input is given by

$$u_k = K\hat{x}_k + \nu_k K e_k \tag{11}$$

where the probability of having access to the full state of the plant is governed by the stochastic variable ν_k, with

$$\nu_k = \begin{cases} 1\ , \text{ with probability } \bar{\nu} \\ 0\ , \text{ with probability } 1-\bar{\nu} \end{cases} \tag{12}$$

and where A, B are the plant matrices, $\hat{A}, \hat{B}$, are the model matrices, u_k is the input, x is the state of the plant, x is the state of the model, and K is the controller. Also, $e_k = x_k - \hat{x}_k$ denotes the error between the plant state and model state.

Notice that the above expressions capture a situation where, if the actual state of the plant is available, that value is used to calculate the control law, but, if it is not available, then the control input is calculated by using the state of the model. Notice also that this corresponds to the case $\tau = 0$, as we might have cases where we have access to the plant for a single clock instant only.

We use the expected total cost as our performance index:

$$J_\infty = \mathbb{E}\left[\sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k\right] \tag{13}$$

We will design an optimal controller K to optimize this performance index. The matrices Q and R are selected by the user according to the performance objective desired, with Q penalizing the system for high values in the state of the plant and R doing so for using excessive control effort.

Our procedure follows that of (24) in that we define a cost-to-go function and calculate it iteratively. To begin, observe that the equations of the system can be rewritten as:

$$\begin{aligned} x_{k+1} &= (A+\nu_k BK)x_k + (1-\nu_k)BK\hat{x}_k \\ \hat{x}_{k+1} &= \nu_k \hat{B}Kx_k + [\hat{A} + (1-\nu_k)\hat{B}K]\hat{x}_k \\ u_k &= \nu_k K x_k + (1-\nu_k)K\hat{x}_k \end{aligned} \tag{14}$$

Let us define $z = \left[x_k^T\ \hat{x}_k^T\right]^T$. We can thus write the equations concerning the plant and model state as $z_{k+1} = F(\nu_k)z$, where:

$$F(\nu_k) = \begin{bmatrix} (A+\nu_k BK) & (1-\nu_k)BK \\ \nu_k \hat{B}K & [\hat{A}+(1-\nu_k)\hat{B}K] \end{bmatrix} \tag{15}$$

The cost-to-go function C_k is defined as follows:

$$C_k^N(z_k) = \mathbb{E}\left[\sum_{h=k}^{N} x_k^T Q x_k + u_k^T R u_k\right] \tag{16}$$

where $Q_k = Q$ and $R_k = R$ except for the terminal cost $R_N = 0$. Following the standard procedure in these cases, we make the claim that this function can be written as

$$C_k^N(z_k) = \mathbb{E}\left[z_k S_k z_k | z_k\right]\ , \tag{17}$$

which is clearly true for $k = N$ and $S_N = Q$. Then, by induction, we show this is true for all k. Suppose it is true for $k+1$, then:

$$\begin{aligned} C_k^N(z_k) &= \mathbb{E}\left[\sum_{h=k}^{N} x_k^T Q x_k + u_k^T R u_k | z_k\right] \qquad (18) \\ &= \mathbb{E}\left[x_k^T Q x_k + u_k^T R u_k + C_{k+1}^N | z_k\right] \\ &= \mathbb{E}\ \begin{array}{c} [x_k^T Q x_k + \\ \begin{pmatrix} \nu_k K x_k + \\ (1-\nu_k)K\hat{x}_k \end{pmatrix}^T R(\nu_k K x_k + (1-\nu_k)K\hat{x}_k) \\ + C_{k+1}^N | z_k] \end{array} \\ &= \mathbb{E}\ \begin{array}{c} [z_k^T \\ \begin{bmatrix} Q + \nu_k^2 K^T RK & (1-\nu_k)\nu_k K^T RK \\ (1-\nu_k)\nu_k K^T RK & (1-\nu_k)^2 K^T RK \end{bmatrix} z_k \\ + z_k^T F^T(\nu_k) S_{k+1} F(\nu_k) z_k | z_k] \end{array} \\ &= \mathbb{E}\ \begin{array}{c} [z_k^T \begin{bmatrix} Q + \bar{\nu}K^T RK & 0 \\ 0 & \bar{\nu}K^T RK \end{bmatrix} z_k \\ + \bar{\nu} z_k^T F^T(0) S_{k+1} F(0) z_k \\ + (1-\bar{\nu}) z_k^T F^T(1) S_{k+1} F(1) z_k | z_k] \end{array} \qquad (19) \end{aligned}$$

Therefore, the above claim is true, and, moreover, we can write that

$$
\begin{aligned}
&S_k && (20)\\
&= \begin{bmatrix} Q+\bar{v}K^TRK & 0 \\ 0 & \bar{v}K^TRK \end{bmatrix} + && (21)\\
&+\bar{v}\begin{bmatrix} A^T & 0 \\ K^TB^T & \hat{A}+K^T\hat{B}^T \end{bmatrix} S_{k+1} \begin{bmatrix} A & BK \\ 0 & \hat{A}+\hat{B}K \end{bmatrix} \\
&+(1-\bar{v})\begin{bmatrix} (A+BK)^T & K^T\hat{B}^T \\ 0 & \hat{A}^T \end{bmatrix} S_{k+1} \begin{bmatrix} A+BK & 0 \\ \hat{B}K & \hat{A} \end{bmatrix} \\
&= \mathscr{F}(S_{k+1},K) \,,
\end{aligned}
$$

where the operator $\mathscr{F}(S_{k+1},K)$ is affine in S for fixed K, and quadratic in K for fixed S.

To obtain the infinite horizon cost, we take the limit as time goes to infinity of the cost-to-go function.

$$J_\infty(K) = \lim_{N\to\infty} C_0^N(x_0) = x_0^T S_\infty x_0 \tag{22}$$

where S_∞ is the solution of the Lyapunov-like equation $S_\infty = \mathscr{F}(S_\infty, K)$, if the solution exists.

Let us now partition the matrix S_∞ as follows

$$S_\infty = \begin{bmatrix} S_1 & S_{12} \\ S_1^T & S_2 \end{bmatrix}. \tag{23}$$

Then the Lyapunov-like equation $S_\infty = \mathscr{F}(S_\infty, K)$ can be expanded as:

$$
\begin{aligned}
S_1 = Q + vK^TRK + vA^TS_1A + \\
(1-v)\begin{bmatrix} (A+BK)^TS_1(A+BK)+ \\ K^T\hat{B}^TS_{12}^T(A+BK)+ \\ (A+BK)^TS_{12}BK + K^T\hat{B}^TS_2BK \end{bmatrix},
\end{aligned} \tag{24}
$$

$$
\begin{aligned}
S_{12} = v\left[A^TS_1BK + A^TS_{12}\hat{A} + A^TS_{12}\hat{B}K\right] + \\
(1-v)\left[(A+BK)^TS_{12}A + K^T\hat{B}_2^TSA\right],
\end{aligned} \tag{25}
$$

$$
\begin{aligned}
S_2 = vK^TRK + \\
v\begin{bmatrix} K^TB^TS_1BK + (\hat{A}^T + K^T\hat{B}^T)S_{12}^TBK \\ +K^TB^TS_{12}(\hat{A}+BK)+ \\ (\hat{A}+\hat{B}K)^TS_2(A+BK) \end{bmatrix} + \\
(1-v)\hat{A}^TS_2\hat{A}.
\end{aligned} \tag{26}
$$

Via algebraic manipulations, we can express all of the above in terms of S_1. Furthermore, the resulting S_1 equation is the only one that depends on the control gain K and can be written as:

$$S_1 = P_1 + P_{12}^TK + K^TP_{12} + K^TP_2K = \mathscr{L}(K, S_1)$$

with P_1, P_{12}, P_2 all linear functions of the matrix S_1 for fixed K. We can also write

$$
\begin{aligned}
\mathscr{L}(K,S_1) &= P_1 - P_{12}^TP_2^{-1}P_{12} \\
&\quad + (K + P_2^{-1}P_{12})^TP_2(K + P_2^{-1}P_{12}) \\
&= \Psi(S_1) + (K - K_S)^TP_2(K - K_S)
\end{aligned}
$$

with

$$\Psi(S_1) = P_1 - P_{12}^TP_2^{-1}P_{12} \tag{27}$$

$$K_S = -P_2^{-1}P_{12} \tag{28}$$

If $P_2 > 0$, then

$$\Psi(S_1) \le \mathscr{L}(K, S_1), \ \forall K$$

where the operator $\Psi(S_1)$ is nonlinear in S_1. The condition $P_2 > 0$ is necessary for stability because, were it not met, we could select K such that it would yield a nonpositive definite S_1, which is not feasible. The previous inequality can be used to find an optimal gain K that minimizes the matrix S_1.

Theorem 3 *Consider the system defined by (10)-(12) and the infinite horizon cost defined in (13). Assume that the pairs (A,B) and $(A^T, Q^{1/2})$ are stabilizable. Then the optimal infinite horizon cost $J_\infty = \min_L J_\infty(K)$ is given by $J_\infty^* = x_0^T T_\infty^* x_0$ where T_∞ is the unique strictly positive solution of the equation:*

$$T_\infty^* = \Psi(T_\infty^*)$$

where $\Psi(T)$ is defined in (6.34) and the optimal gain is given by

$$K^* = K_{T_\infty^*}$$

with K_S as defined in (28). The equation $T_\infty^* = \Psi(T_\infty^*)$ has a positive definite solution if and only if $\bar{v} > v_c$, where v_c is a critical probability of having access to the channel, which depends on the pairs (A,B) and $(\hat{A},\hat{B})$. The T_∞^* can be obtained as the limit of the sequence $T_{k+1} = \Psi(T_k)$, that is, $\lim_{k\to\infty} T_k = T_\infty$.

4. Conclusions and future work

In this paper, we investigated the behavior of the system while following a reference input. We provided the full response of the system and a stability condition. We then designed an optimal controller for MB-NCS to meet an LQ-like performance index, by using dynamic programming techniques. Some extensions of these results can be found in [9] and will continue to be developed in future work.

Additionally, we will seek to use intermittent feedback to improve performance, by updating the model during the times when the system is running closed loop, with the aim of enabling the user to run the system closed loop for progressively shorter intervals.

References

[1] P. Antsaklis and A. Michel, *Linear Systems*, 1st edition, McGraw-Hill, New York, 1997.

[2] J. Baillieul, P. Antsaklis. ”Control and communication challenges in networked real-time systems,” Proceedings of the IEEE, v 95, n 1, Jan. 2007, p 9-28.

[3] B. Azimi-Sadjadi, ”Stability of Networked Control Systems in the Presence of Packet Losses,” Proceedings of the 42nd Conference of Decision and Control, December 2003, pp. 676- 681.

[4] M.S. Branicky, S. Phillips, and W. Zhang, ”Scheduling and feedback co-design for networked control systems,” Proceedings of the 41st Conference on Decision and Control, December 2002, pp. 1211- 1217.

[5] D. Delchamps, ”Stabilizing a Linear System with Quantized State Feedback,” IEEE Transactions on Automatic Control, vol. 35, no. 8, pp. 916-924.

[6] T. Estrada, H. Lin, and P.J. Antsaklis, ”Model-Based Control with Intermittent Feedback”, Proceedings of the 14th Mediterranean Conference on Control and Automation, 2006, pp. 1-6.

[7] T. Estrada and P.J. Antsaklis, ”Control with Intermittent Sensor Measurements: A New Look at Feedback Conrol”, Proceedings of the Workshop on Networked Distributed Systems for Intelligent Sensing and Control, Athens, Greece, 2007.

[8] T. Estrada and P.J. Antsaklis, ”Results on Continuous and Discrete Model-Based Networked Control Systems with Intermittent Feedback, Part I: Stability.”, ISIS Technical Report ISIS-2008-001, April 2008.

[9] T. Estrada and P.J. Antsaklis, ”Results on Continuous and Discrete Model-Based Networked Control Systems with Intermittent Feedback, Part II: Performance.”, ISIS Technical Report, April 2009.

[10] E. Fridman, A. Seuret, J.P. Richard, “Robust sampled-data stabilization of linear systems: an input delay approach”, Automatica, 2004.

[11] J. Hespanha, P. Naghshtabrizi, Yonggang Xu, ”A survey of recent results in networked control systems”, Proceedings of the IEEE, v 95, n 1, Jan. 2007, p 138-62.

[12] D. Hristu-Varsakelis, ”Feedback Control Systems as Users of a Shared Network: Communication Sequences that Guarantee Stability,” Proceedings of the 40th Conference on Decision and Control, December 2001, pp. 3631-36.

[13] M. Kim, et al, ”Controlling chemical turbulence by global delayed feedback:Pattern formation in catalytic co oxidation” on pt(110), Science, vol. 292, no. 5520, 2001.

[14] K. Koay and G. Bugmann, ”Compensating intermittent delayed visual feedback in robot navigation”, Proceedings of the IEEE Conference on Decision and Control, 2004.

[15] D. Liberzon and R. Brockett, ”Quantized Feedback Stabilization of Linear Systems,” IEEE Transactions on Automatic Control, Vol 45, no 7, pp 1279-89, July 2000.

[16] L.A. Montestruque, PhD Dissertation, Model-Based Networked Control Systems, University of Notre Dame, Notre Dame IN, USA, September 2004.

[17] L.A. Montestruque and P.J. Antsaklis, ”Model-Based Networked Control Systems: Necessary and Sufficient Conditions for Stability”, 10th Mediterranean Conference on Control and Automation, July 2002.

[18] L.A. Montestruque and P.J. Antsaklis, ”State and Output Feedback Control in Model-Based Networked Control Systems”, 41st IEEE Conference on Decision and Control, December 2002, pp. 1620- 1625.

[19] L.A. Montestruque and P.J. Antsaklis, ”On the model-based control of networked systems,” Automatica, Vol. 39, pp 1837-1843.

[20] L.A. Montestruque and P.J. Antsaklis, ”Networked Control Systems: A model-based approach,” 12th Mediterranean Conference on Control and Automation, June 2004.

[21] E. Ronco and D. J. Hill, ”Open-loop intermittent feedback optimal predictive control:a human movement control model”, NIPS99, 1999.

[22] G. Nair and R. Evans, ”Communication-Limited Stabilization of Linear Systems,” Proceedings of the Conference on Decision and Control, 2000, pp. 1005-1010.

[23] B. H. Salzberg, A. J. Wheeler, L. T. Devar, and B. L. Hopkins, ”The effect of intermittent feedback and intermittent contingent access to play on printing of kindergarten children”, J Anal Behav, vol. 4, no. 3, 1971.

[24] L. Schenato. “To zero or to hold control inputs in lossy networked control systems”, Proceedings of the European Control Conference, Kos, Greece, 2007.

[25] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. “Foundations of Control and Estimation over Lossy Networks.“ Proceedings of the IEEE, Special Issue on Networked Control Systems. Vol. 95, Issue 1, January 2007.

[26] Richard A. Schmidt, *Motor Control and Learning - A Behaviorial Emphasis*. Human Kinetics, 4th ed., 2005.

[27] J. Took, D. Tilbury, and N. Soparkar, ”Trading Computation for Bandwidth: Reducing Computation in Distributed Control Systems using State Estimators,” IEEE Transactions on Control Systems Technology, July 2002, Vol 10, No 4, pp 503-518.

[28] G. Walsh, H. Ye, and L. Bushnell, ”Stability Analysis of Networked Control Systems,” Proceedings of American Control Conference, June 1999.