Model-Based Control of Continuous-Time and Discrete-Time Systems with Large Network Induced Delays.

Eloy Garcia and Panos J. Antsaklis, *Fellow, IEEE*

Abstract—**Stabilization of continuous and discrete-time systems in the presence of network induced delays and model uncertainties is studied in this paper using the Model-Based Networked Control Systems (MB-NCS) framework. The use of a nominal model of the system to generate an estimate of the real state between measurement update intervals allows for significant reduction of traffic in the network. The work in this paper extends previous results in MB-NCS that dealt with continuous-time systems and small delays. In the current paper we are able to obtain necessary and sufficient conditions for stability for the case of large delays, that is, when the network delays are larger than the update intervals. Additionally, similar conditions are derived for discrete-time systems and for small and large delays.**

I. Introduction

In Networked Control Systems (NCS) a digital communication network is used to transfer information among the components of a control system. NCS can also improve efficiency, flexibility, and reliability of the network interconnected system reducing reconfiguration and maintenance costs [1]. In contrast, the protocols used to establish an ordered communication between nodes in the network introduce time delays and loss of information. Network induced delays represent an important problem in the design and analysis of NCS since traditional control techniques may be sensitive to the large delays that are present in networked applications. Time delays due to network interconnections can occur for different reasons such as: pre- and post-processing times, which are the times that are needed to measure, encode, and decode data; waiting time, which is the elapsed time from when a packet of information is ready to be transmitted until the network allows its transmission; and transmission time, which is the time needed for the data to be transmitted from their source node to their destination node [2]-[3]. Time delays in control systems are not only due to network interconnections but they arise naturally in different areas; see [4] for details and control techniques.

There exist different approaches that consider the presence of network induced delays [5]-[10]. For instance, the work in [5] presents a configuration that stabilizes a NCS with large constant delays using passivity and the scattering transformation. The works in [6] and [7] derive general models of NCSs that consider time-varying sampling intervals and delays.

Both authors are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: egarcia7@nd.edu). The support of the National Science Foundation under Grant No. CNS-1035655 is gratefully acknowledged.

The authors of the previous papers do not consider model uncertainties. In contrast, we are able to provide robustness to parameter uncertainties in the presence of delays by following the Model-Based Networked Control Systems MB-NCS framework. Network induced delays and model uncertainties are also considered in [11]. In this work the total delay is restricted to be less than the sampling period. A Zero-Order-Hold (ZOH) controller is used which typically results in small sampling periods.

In the present paper we use the nominal model of the system to estimate the current plant state based on the delayed measurements. The control input is now a function of the model state which, depending on the model uncertainties, can provide better system performance and allow for longer sampling intervals than using the delayed state directly to compute the control input. The Smith Predictor has been commonly used for prediction of the current output of a system when only delayed measurements are available [12]. In this case it is necessary for the predictor to know the exact parameters of the system and to receive continuous measurements in order to obtain accurate estimates of the current system output. In the present paper we relax those constraints by using a nominal, inexact model of the system and using non-continuous measurements.

The paper is organized as follows: section II provides brief background on the MB-NCS framework. Section III presents stability conditions for discrete-time systems and for the small delay case. Section IV presents our main results, necessary and sufficient conditions for stability for the large delay case for both continuous-time and discrete-time systems. An illustrative example is offered in section V and section VI summarizes the results of this paper.

II. Background on MB-NCS

MB-NCS were introduced in [13]; this configuration makes use of an explicit model of the plant which is added to the actuator/controller node to compute the control input based on the state of the model rather than on the plant state. The state of the model is updated when the controller receives the measured state of the plant that is sent from the sensor node every h time units. Fig. 1 shows the interconnection of several NCSs. The labeled small blocks correspond to each system's actuator and sensor nodes. The actuator/controller node in MB-NCS with delays can be represented as in Fig. 2. We assume that the systems are decoupled, i.e. the dynamics of each system in Fig. 1 depend only on its own state. Without loss of generality we will focus on a particular system/model pair. The dynamics of the plant and the model can be described respectively by:

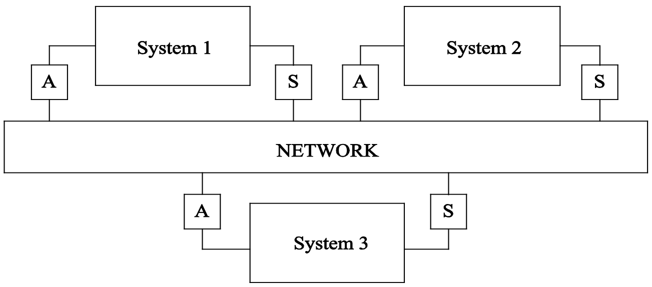

Fig. 1. Networked Control Systems.

$$\dot{x} = Ax + Bu \quad (1)$$

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \quad (2)$$

where $x, \hat{x} \in \mathbb{R}^n$, $u = K\hat{x}$, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices *A,B*.

The plant may be unstable i.e. not all eigenvalues of *A* have negative real parts. In [15] the authors provided necessary and sufficient conditions for stability of continuous-time systems when the updates from the system are periodic (every *h* time units) for the case when the delays are negligible and also when there exist small constant delays $\tau<h$. See sections III and IV for further details about delays, update intervals, and their relationship for the small and large delay cases. Other results based on the MB-NCS framework that consider network induced delays can be found in [16]-[17], but they consider only the small delay case as well. In this paper we extend this framework in order to consider discrete time systems and, more importantly, to consider large delays, $\tau>h$, for both continuous and discrete time systems.

III. Discrete-time systems with small delays

In this section we consider multi-input, multi-output linear time-invariant discrete-time systems and models. In this case the updates will occur at some of the discrete time instants indexed by *n* that correspond to the operation and sampling of the plant. This implies that the update interval *h* will be an integer number, representing the number of plant discrete instants between two consecutive measurements that the sensor broadcasts. Let n_k represent the update instants, then $h = n_{k+1} - n_k$.

We call a 'MB-NCS with small delays' the case when the update network interval *h* is greater than the network induced delays. Note that the delay can be greater than the sampling time of the system indexed by *n* as shown in Fig. 3. Therefore, in our approach, a small delay is greater than the sampling time of the plant but smaller than the update interval *h* at which the sensor decides to send information to update the model. For discrete-time systems we assume that the update time *h* is constant and an integer number. We also assume the delay τ is constant and an integer number; it represents the number of system sampling instants that the information is delayed. We will present here the case of full state feedback systems. For this case we have that at times $n_k - \tau$ the sensor transmits the state data to the controller/ actuator. This data will arrive τ plant samplings later. So, at

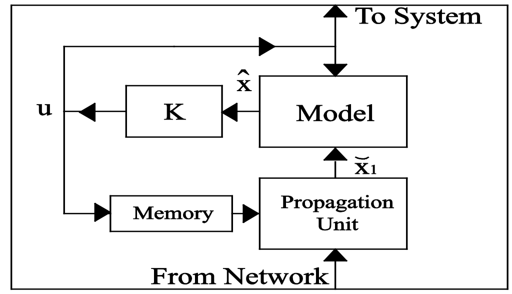

Fig. 2. Model-Based Networked Control System actuator/controller node with propagation unit.

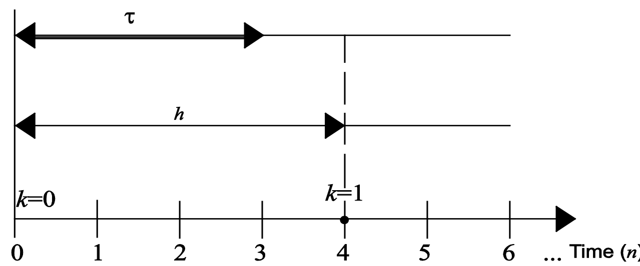

Fig. 3. Representation of the small delay case and the involved parameters for discrete-time systems. In this example h=4 and τ=3<h.

times *kh, k*=1,2,... the controller/actuator receives the state vector value $x(n_k - \tau)$. The main idea is to use the model in the controller to estimate the present value of the real state and use this estimated state to update the controller's model.

The Propagation Unit uses the plant model and the past values of the control input $u(n)$ to calculate an estimate $\breve{x}(n_k)$ of the current state $x(n_k)$ from the received data $x(n_k - \tau)$. This estimate is then used to update the model that with the controller will generate the control signal for the plant. Fig. 2 represents the configuration of the components in the actuator node.

The components of the control system are described by the following equations:

$$\begin{array}{ll} \text{Plant:} & x(n+1) = Ax(n) + Bu(n) \\ \text{Model:} & \hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n),\ n \in [n_k, n_{k+1}) \\ \text{Controller:} & u(n) = K\hat{x}(n) \\ \text{Propagation} & \\ \text{Unit:} & \breve{x}(n+1) = \hat{A}\breve{x}(n) + \hat{B}u(n),\ n \in [n_k - \tau, n_{k+1} - \tau) \end{array} \quad (3)$$

The following update law is used in order to find the admissible delays and update intervals:

$$\text{Update law:} \quad \left\{ \begin{array}{ll} x \rightarrow \breve{x}, & n = n_k - \tau \\ \breve{x} \rightarrow \hat{x}, & n = n_k \end{array} \right\}. \quad (4)$$

In order to analyze the stability properties of system (3) using update law (4), we initialize the propagation unit at time n_k - τ with the state vector that the sensor obtains. The model and propagation unit operate together until n_k. At this time, the model is updated with the propagation unit state vector, as described in (4). This is equivalent to having the propagation unit receive the state vector $x(n_k$ - $\tau)$ at n_k and to instantaneously compute an estimate $\breve{x}(n_k)$ of the current state $x(n_k)$. Note that the plant represents a physical system to be controlled and its state cannot be updated externally. The measured plant states are used to update the propagation unit. Both, model and propagation unit are implemented as computing algorithms in the controller node and their states can be accessed and updated at any time as necessary.

We define the errors $\hat{e}(n) = \breve{x}(n) - \hat{x}(n)$ and $\breve{e}(n) = x(n) - \breve{x}(n)$. It can be shown that the dynamics of the state and the errors can be represented by:

$$\begin{aligned} x(n+1) &= (A+BK)x(n) - BK\breve{e}(n) - BK\hat{e}(n) \\ \breve{e}(n+1) &= (\tilde{A}+\tilde{B}K)x(n) + (\hat{A}-\tilde{B}K)\breve{e}(n) - \tilde{B}K\hat{e}(n) \\ \hat{e}(n+1) &= \hat{A}\hat{e}(n) \end{aligned} \quad (5)$$

where $\tilde{A} = A - \hat{A},\ \tilde{B} = B - \hat{B}.$ Define the augmented state vector $z = [x^T\ \breve{e}^T\ \hat{e}^T]^T$. Then the augmented system can be represented in compact form as:

$$z(n+1) = \Lambda z(n), \quad (6)$$

where $\Lambda = \begin{bmatrix} A+BK & -BK & -BK \\ \tilde{A}+\tilde{B}K & \hat{A}-\tilde{B}K & -\tilde{B}K \\ 0 & 0 & \hat{A} \end{bmatrix}$. According to the update laws (4) we have the augmented state reset equations:

$$z(n_k-\tau) = \begin{bmatrix} x((n_k-\tau)^-) \\ 0 \\ \breve{e}((n_k-\tau)^-) + \hat{e}((n_k-\tau)^-) \end{bmatrix}, \quad z(n_k) = \begin{bmatrix} x(n_k^-) \\ \breve{e}(n_k^-) \\ 0 \end{bmatrix} \quad (7)$$

where $n_{k+1} - n_k = h, \quad 0 < \tau < h.$

The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described as shown next.

Proposition 1. The system with dynamics described by (3)-(4) with initial conditions $z(n_0) = [x_0^T\ \breve{e}_0^T\ \hat{e}_0^T]^T = z_0$, $n_0 = 0,$ has the following response:

$$z(n) = \Lambda^{(n-n_k)}\Sigma^k z_0 \quad \text{for } n \in [n_k, n_{k+1}-\tau) \quad (8)$$

$$z(n) = \Lambda^{(n-n_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)}\Sigma^k z_0 \quad \text{for } n \in [n_{k+1}-\tau, n_{k+1}).$$

where $\Sigma = \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} \right).$

Proof. Assume, without loss of generality, that the system starts at time $n_0 = 0$ with initial conditions $z(n_0) = [x_0^T\ \breve{e}_0^T\ \hat{e}_0^T]^T = z_0$. On the interval $n \in [0, n_1 - \tau)$, the system response is:

$$z(n) = \begin{bmatrix} x(n) \\ \breve{e}(n) \\ \hat{e}(n) \end{bmatrix} = \Lambda^n z_0. \quad (9)$$

At $n = (n_1 - \tau)^-$ the state of the system is given by:

$$z((n_1-\tau)^-) = \Lambda^{(n_1-\tau)} z_0 = \Lambda^{(h-\tau)} z_0. \quad (10)$$

At time $n = (n_1 - \tau)$ we update according to (4), $x \to \breve{x}$, $\breve{e}(n_1-\tau) = 0$ and $\hat{e}(n_1-\tau) = \breve{e}(n_1-\tau)^- + \hat{e}(n_1-\tau)^-$, we have:

$$z(n_1-\tau) = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} z_0. \quad (11)$$

Continuing with the interval $n \in [n_1-\tau, n_1)$, we have at $n = n_1^-$:

$$z(n_1^-) = \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} z_0. \quad (12)$$

Similarly, at time $n = n_1$ we use the update law (4), that means $\hat{e}(n_1) = 0$ and the state is given by:

$$z(n_1) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} z_0 = \Sigma z_0. \quad (13)$$

Following the same analysis at every cycle $n \in [n_k, n_{k+1}]$ we obtain the response of the system given by (8). ■

Theorem 2. The networked system described by (3)-(4) with constant updates *h* and constant delays *τ<h* is globally exponentially stable around the solution $z_0 = 0$ if and only if the eigenvalues of

$$\Sigma = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} \quad (14)$$

are inside the unit circle. ■

Remark 1. The analysis presented above for discrete-time systems offers an important advantage compared to its continuous-time counterpart [15]. To compute propagated states based on delayed measurements we need to store the previous values of the control input that were used over the interval $[n_k - \tau, n_k)$. For discrete-time systems the control input history over $[n_k - \tau, n_k)$ can be represented by a finite number of values but for continuous-time systems it is not possible to store in digital memory an infinite number of values that characterize the input *u*(*t*). In this case we need to sample sufficiently fast in order to obtain a good approximation of the continuous control input *u*(*t*).

IV. MB-NCS WITH LARGE DELAYS

A. *Continuous-time systems with large delays.*

In this section we extend the approach discussed in the previous section to consider the case when *τ>h*, for both continuous and discrete-time systems. The aim is to obtain conditions for stability in the presence of delays that are larger than the periodic update intervals. The solution to this problem is obtained by considering an increased number of propagation state variables which, in turn, requires the state vector *z* to be augmented to include additional error variables. For continuous time systems *h* and *τ* can be real numbers, they are not restricted to be integers. We consider first for simplicity the case when *h<τ<2h*, the general case *ah<τ<(a+1)h* for any positive integer *a* can be solved using the same approach but adding more error variables.

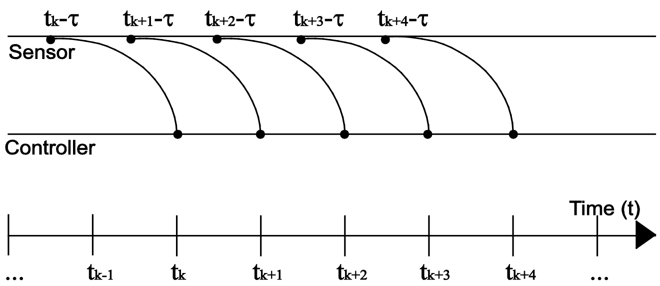

Fig 4. Representation of update intervals in the presence of delays $h<\tau<2h$ for continuous-time systems.

We consider two propagation state variables $\breve{x}_1$ and $\breve{x}_2$. We update the state $\breve{x}_2$ at time $t_k-\tau$ using the real state $x(t_k-\tau)$. At time t_k we perform two updates in sequential manner. First, we update the state of the model $\hat{x}(t_k)$ using the value $\breve{x}_1(t_k)$ and then we update $\breve{x}_1(t_k)$ using the value $\breve{x}_2(t_k)$. The overall setup for continuous-time systems can be represented in compact form as follows:

$$\begin{array}{ll} \text{Plant:} & \dot{x}(t)=Ax(t)+Bu(t) \\ \text{Model:} & \dot{\hat{x}}(t)=\hat{A}\hat{x}(t)+\hat{B}u(t),\ t\in[t_k,t_{k+1}) \\ \text{Controller:} & u(t)=K\hat{x}(t) \\ \text{Propagation} & \dot{\breve{x}}_1(t)=\hat{A}\breve{x}_1(t)+\hat{B}u(t),\ t\in[t_k,t_{k+1}) \\ \text{Units:} & \dot{\breve{x}}_2(t)=\hat{A}\breve{x}_2(t)+\hat{B}u(t),\ t\in[t_k-\tau,t_{k+1}-\tau) \end{array} \tag{15}$$

$$\text{Update law:}\left\{\begin{array}{c} x\rightarrow\breve{x}_2,\quad t=t_k-\tau \\ \breve{x}_1\rightarrow\hat{x},\ \text{then}\ \breve{x}_2\rightarrow\breve{x}_1,\quad t=t_k \end{array}\right\}. \tag{16}$$

Using this representation we ensure that the model of the system that generates the state $\hat{x}(t)$ is updated at time t_k with the propagated variable that is computed based on the measurement that was sent by the sensor at time $t_k-\tau$, see Fig. 4. In this figure we can see that at time t_k the model in the controller is updated with information generated at time $t_k-\tau$. Even though a new measurement has been sent by the senor node at time $t_{k+1}-\tau$, this new information has not arrived yet to the controller node due to large delay τ.

Remark 2. Note that in practice we only need one propagation unit that receives the delayed measurement and propagates it instantaneously to time t_k using the previous control input $u(t)$ stored over the interval $[t_k-\tau,t_k]$. Therefore, the implementation is as shown in Fig. 2.

Define the errors $\breve{e}(t)=x(t)-\breve{x}_2(t)$, $\hat{e}(t)=\breve{x}_1(t)-\hat{x}(t)$ and $\overset{\frown}{e}(t)=\breve{x}_2(t)-\breve{x}_1(t)$. It can be shown that the dynamics of the state and the errors can be represented by:

$$\begin{aligned} \dot{x}(t)&=(A+BK)x(t)-BK\breve{e}(t)-BK\hat{e}(t)-BK\overset{\frown}{e}(t) \\ \dot{\breve{e}}(t)&=(\tilde{A}+\tilde{B}K)x(t)+(\hat{A}-\tilde{B}K)\breve{e}(t)-\tilde{B}K\hat{e}(t)-\tilde{B}K\overset{\frown}{e}(t) \\ \dot{\hat{e}}(t)&=\hat{A}\hat{e}(t) \\ \dot{\overset{\frown}{e}}(t)&=\hat{A}\overset{\frown}{e}(t). \end{aligned} \tag{17}$$

Define the augmented state vector $z=[x^T\ \breve{e}^T\ \hat{e}^T\ \overset{\frown}{e}^T]^T$ then the augmented system can be represented in compact form:

$$\dot{z}(t)=\Lambda z(t), \tag{18}$$

where $\Lambda=\begin{bmatrix} A+BK & -BK & -BK & -BK \\ \tilde{A}+\tilde{B}K & \hat{A}-\tilde{B}K & -\tilde{B}K & -\tilde{B}K \\ 0 & 0 & \hat{A} & 0 \\ 0 & 0 & 0 & \hat{A} \end{bmatrix}$.

From (16) we obtain the augmented state reset equations:

$$z(t_k-\tau)=\begin{bmatrix} x((t_k-\tau)^-) \\ 0 \\ \hat{e}((t_k-\tau)^-) \\ \breve{e}((t_k-\tau)^-)+\overset{\frown}{e}((t_k-\tau)^-) \end{bmatrix},\quad z(t_k)=\begin{bmatrix} x(t_k^-) \\ \breve{e}(t_k^-) \\ \overset{\frown}{e}(t_k^-) \\ 0 \end{bmatrix} \tag{19}$$

where $t_{k+1}-t_k=h,\quad h<\tau<2h.$

Proposition 3. The system with dynamics described by (15)-(16) with initial conditions $z(t_0)=[x_0^T\ \breve{e}_0^T\ \hat{e}_0^T\ \overset{\frown}{e}_0^T]^T=z_0$, $t_0=0$, has the following response:

$$z(t)=e^{\Lambda(t-t_k)}\Sigma^k z_0\quad \text{for } t\in[t_k,t_{k+1}-\tau') \tag{20}$$

$$z(t)=e^{\Lambda(t-t_{k+1}+\tau')}\begin{bmatrix} I&0&0&0 \\ 0&0&0&0 \\ 0&0&I&0 \\ 0&I&0&I \end{bmatrix}e^{\Lambda(h-\tau')}\Sigma^k z_0\quad \text{for } t\in[t_{k+1}-\tau',t_{k+1})$$

where $\Sigma=\begin{bmatrix} I&0&0&0 \\ 0&I&0&0 \\ 0&0&0&I \\ 0&0&0&0 \end{bmatrix}e^{\Lambda\tau'}\begin{bmatrix} I&0&0&0 \\ 0&0&0&0 \\ 0&0&I&0 \\ 0&I&0&I \end{bmatrix}e^{\Lambda(h-\tau')}$, and $\tau'=\tau-h$.

Proof. Following a similar analysis as in the proof of Proposition 1 we have that on the interval $t\in[0,t_1-\tau')$ and for initial time $t_0=0$ with initial conditions $z(t_0)=[x_0^T\ \breve{e}_0^T\ \hat{e}_0^T\ \overset{\frown}{e}_0^T]^T=z_0$, the system response is

$$z(t)=\begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \\ \overset{\frown}{e}(t) \end{bmatrix}=e^{\Lambda t}z_0. \tag{21}$$

At $t=(t_1-\tau')^-$ the state of the system is given by:

$$z((t_1-\tau')^-)=e^{\Lambda(t_1-\tau')}z_0=e^{\Lambda(h-\tau')}z_0. \tag{22}$$

At time $t=(t_1-\tau')$ we update according to (16), $x\rightarrow\breve{x}_2$, which means that $\breve{e}(t_1-\tau')=0$ and $\overset{\frown}{e}(t_1-\tau')=\breve{e}(t_1-\tau')^-+\overset{\frown}{e}(t_1-\tau')^-$, then we have:

$$z(t_1-\tau')=\begin{bmatrix} I&0&0&0 \\ 0&0&0&0 \\ 0&0&I&0 \\ 0&I&0&I \end{bmatrix}e^{\Lambda(h-\tau')}z_0. \tag{23}$$

Continuing with the interval $t\in[t_1-\tau',t_1)$, at $t=t_1^-$:

$$z(t_1^-)=e^{\Lambda\tau'}\begin{bmatrix} I&0&0&0 \\ 0&0&0&0 \\ 0&0&I&0 \\ 0&I&0&I \end{bmatrix}e^{\Lambda(h-\tau')}z_0. \tag{24}$$

Similarly, at time $t = t_1$ we update according to (16), that means $\overset{\frown}{e}(t_1) = 0$ and $\hat{e}(t_1) = \overset{\frown}{e}(t_1^-)$, then the state is given by:

$$z(t_1) = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')} z_0. \quad (25).$$

By following the same analysis at every cycle we obtain the response of the system described in (20).■

Theorem 4. The networked system described by (15)-(16) with constant updates h and constant delays $h<\tau<2h$ is globally exponentially stable around the solution $z_0 = 0$ if and only if the eigenvalues of

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')} \quad (26)$$

are inside the unit circle. ■

Remark 3. The MB-NCS framework provides a range of possible values for the update interval h in order to obtain a stable control system. The work in [15] provided stability conditions only when $\tau<h$. The most important advantage of the results presented in this section can be better appreciated in the next scenario: for a given plant, model, controller, and network delay there may not be an update interval $h>\tau$ that stabilizes the system, but using the results in this section, we can find some $h<\tau$ that results in a stable system.

We can extend the previous results in order to establish necessary and sufficient conditions for stability for the general case when $ah<\tau<(a+1)h$ by adding a additional propagation state variables with respect to the small delay case described in section III. For instance, in the above analysis we had that $h<\tau<2h$ so $a=1$ and we added one additional propagation variable with respect to the small delay case and we had a total of two propagation variables.

Theorem 5. The networked system described by (15)-(16) with constant updates h, constant delays $ah<\tau<(a+1)h$, and with augmented state $z = [x^T\ \breve{e}^T\ \hat{e}^T\ \overset{\frown}{e}_1^T \ldots \overset{\frown}{e}_a^T]^T$ is globally exponentially stable around the solution $z_0 = 0$ if and only if the eigenvalues of

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & I \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')} \quad (27)$$

are inside the unit circle, where $\tau' = \tau - ah$, $\breve{e}(t) = x(t) - \breve{x}_{a+1}(t)$, $\hat{e}(t) = \breve{x}_1(t) - \hat{x}(t)$, $\overset{\frown}{e}_i(t) = \breve{x}_{i+1}(t) - \breve{x}_i(t)$, for i=1,2 ...a, and the $(a+3)\times(a+3)$ matrix Λ is given by:

$$\Lambda = \begin{bmatrix} A+BK & -BK & -BK & -BK & \ldots & -BK \\ \tilde{A}+\tilde{B}K & \hat{A}-\tilde{B}K & -\tilde{B}K & -\tilde{B}K & \ldots & -\tilde{B}K \\ 0 & 0 & \hat{A} & 0 & \ldots & 0 \\ 0 & 0 & 0 & \hat{A} & \ldots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \hat{A} \end{bmatrix}. \ ■ \quad (28)$$

B. Discrete-time systems with large delays.

We follow the same constraint as in section III, that is, h and τ can take only integer values, but in contrast to section III we now can consider the case $\tau>h$.

Theorem 6. The networked system described by (3)-(4) with constant updates h, constant delays $ah<\tau<(a+1)h$, and with augmented state $z = [x^T\ \breve{e}^T\ \hat{e}^T\ \overset{\frown}{e}_1^T \ldots \overset{\frown}{e}_a^T]^T$ is globally exponentially stable around the solution $z_0 = 0$ if and only if the eigenvalues of

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & I \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \Lambda^{\tau'} \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & I \end{bmatrix} \Lambda^{(h-\tau')} \quad (29)$$

are inside the unit circle, where $\tau' = \tau - ah$, $\breve{e}(n) = x(n) - \breve{x}_{a+1}(n)$, $\hat{e}(n) = \breve{x}_1(n) - \hat{x}(n)$, $\overset{\frown}{e}_i(n) = \breve{x}_{i+1}(n) - \breve{x}_i(n)$, for i=1,2 ...a, and the $(a+3)\times(a+3)$ matrix Λ is given by (28). ■

V. Example

Consider the following unstable continuous-time system implemented as in Fig. 1 and using a model and propagation unit as represented in Fig. 2:

$$A = \begin{bmatrix} -0.349 & 0.65 \\ -0.316 & 0.787 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \hat{A} = \begin{bmatrix} -0.5 & 1 \\ 0 & 0.6 \end{bmatrix}, \hat{B} = \begin{bmatrix} -0.008 \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} -5.4621 & -11.1658 \end{bmatrix}.$$

Suppose that the networked induced delay is constant and equal to 1.6 seconds. For this system, model, and delay there is no update interval $h>\tau$ that provides stability as shown in the top part of Fig. 5. For h>1.6 seconds the corresponding matrix Σ has at least one eigenvalue with magnitude greater than one. If we decrease the update interval as shown in the bottom part of Fig. 5, there exist values of h for which the system is stable. The corresponding matrix Σ for each plot can be obtained from (28) using the appropriate number of propagation variables. An example of a stable response of the system for τ=1.6 seconds is shown in Fig. 7.a. using h=1.2 seconds.

Suppose now that the delay is larger, τ=1.85 seconds. Fig. 6 shows that stabilizing values of h exist for the range for $2h<\tau<3h$ (c), but not for $h<\tau<2h$ (b) or for $\tau<h$ (a). The response of the system for τ=1.85 seconds and using h=0.73 seconds is shown in Fig. 7.b.

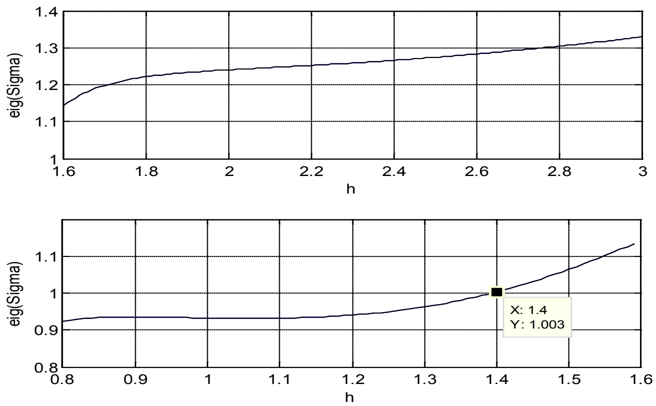

Fig. 5. τ=1.6 seconds. Eigenvalue of Σ with largest magnitude for τ<h (top) and for h<τ<$2h$ (bottom).

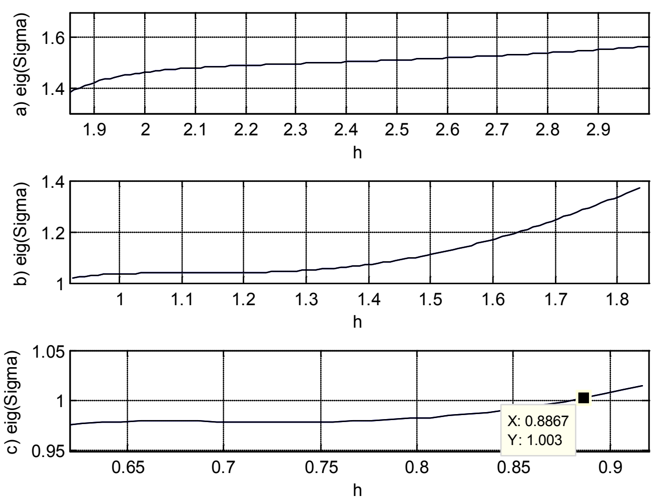

Fig. 6. τ=1.85 seconds. Eigenvalue of Σ with largest magnitude for τ<h (a), for h<τ<$2h$ (b), and for $2h$<τ<$3h$ (c).

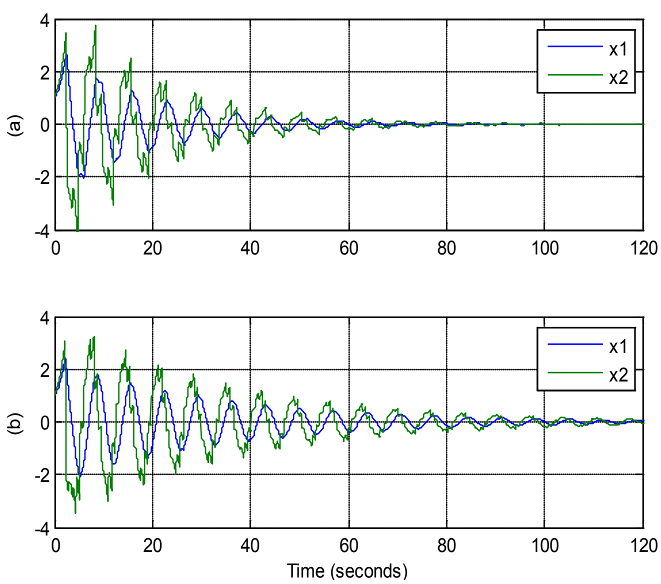

Fig. 7. Response of networked system in example 1. (a) For τ=1.6 seconds and h=1.2 seconds. (b) For τ=1.85 seconds and h=0.73 seconds.

VI. Conclusion

The results in this paper represent an important extension to the framework of MB-NCS with delays. We analyzed the case when τ>h and presented necessary and sufficient conditions for stability that depend on the system and model dynamics but also on the values of the delay and the update interval. We can obtain a range of h for stability following a similar approach than in [15] that uses the state of the model for control and a propagation unit that provides an estimate of the real state using delayed measurements and previous input values. The main advantage of the analysis presented here is that, when a stabilizing $h>\tau$ does not exist, it is still possible to obtain update intervals $h<\tau$ that stabilize the system. The results in this paper provide stability results for MB-NCS with delays and relaxing the constraint that the delay has to be always less than the update interval.

References

[1] P. J. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.

[2] F. L. Lian, J. R. Moyne, D. M. Tilbury "Performance evaluation of control networks: Ethernet, Controlnet, and Devicenet," *IEEE Control Systems Magazine*, pp. 66 – 83, February 2001.

[3] J. M. Moyne, D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29 – 47, January 2007.

[4] R. Sipahi, S. I. Niculescu, C. T. Abdallah, W. Michiels, ans K. Gu, "Stability and stabilization of systems with delays: limitations and opportunities," *IEEE Control Systems Magazine*, vol. 31, no.1, pp. 38 – 65, February 2011.

[5] T. Matiakis, S. Hirche, and M. Buss, "Independent-of-delay stability of nonlinear networked control systems by scattering transformation," *in Proc. American Control Conference*, 2006, pp. 2801 – 2806.

[6] M. B. G. Cloosterman, L. Hetel, N. van de Wouw, W. P. M. H.Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked systems," *Automatica*, vol. 46, pp. 1584 – 1594, 2010.

[7] L. Hetel, J. Daafouz, and C. Iung, "Analysis and control of LTI and switched systems in digital loops via an event-based modeling," *International Journal of Control*, vol. 81, no.7, pp. 1125 – 1138, 2008.

[8] H. Gao, T. Chen, and J. Lam, "A new delay system approach to network based control," *Automatica*, vol. 44, no. 1, pp. 39 – 52, January 2008.

[9] M. S. Branicky, S. M. Phillips, and W. Zhang, "Stability of networked control systems: explicit analysis of delys," *in Proc. American Control Conference*, 2000, pp. 2352 – 2357.

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

[11] A. H. Tahoun and H. Fang, "Adaptive stabilization of networked control systems with delays and state-dependent disturbances," in *Proc. of 17th Mediterranean Conference on Control and Automation*, 2009, pp. 622 – 627.

[12] Z. J. Palmor, "Time-delay compensation-Smith predictor and its modifications," *The Control Handbook,* S. Levine, Ed. New York: CRC and IEEE, 1996.

[13] L. A. Montestruque and P. J. Antsaklis, "Model-based networked control systems: necessary and sufficient conditions for stability" *Proc. of the 10th Mediterranean Conference on Control and Automation*, 2002.

[14] L. A. Montestruque and P. J. Antsaklis. "State and output feedback control in model-based networked control systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[15] L. A. Montestruque, P. J. Antsaklis "On the model-based control of networked systems," *Automatica*, vol. 39, no. 10, pp. 1837 – 1843, 2003.

[16] S. Mu, T. Chu, and L. Wang, "Impulsive control of networked systems with communication delays," *in Proc. American Control Conference*, 2005, pp. 650 – 655.

[17] Z. Wang, W. Liu, H. Dai, D. S. Naidu, "Robust stabilization of model-based uncertain singularly perturbed systems with network time-delay," *in Proc. of 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 7917 – 7922.