Luis A. Montestruque and Panos J. Antsaklis, "Model-Based Networked Control Systems - Stability," ISIS Technical Report ISIS-2002-001, January 2002.

MODEL-BASED NETWORKED CONTROL SYSTEMS – STABILITY

Luis A. Montestruque, Panos J. Antsaklis
University of Notre Dame, Notre Dame, IN 46556
Department of Electrical Engineering

1. ABSTRACT

In this report the control of continuous linear plants where the state sensor is connected to a linear controller/actuator via a network is addressed. The work focuses on reducing the network usage using knowledge of the plant dynamics. Specifically, the controller uses an explicit model of the plant that approximates the plant dynamics and makes possible stabilization of the plant even under slow network conditions. Necessary and sufficient conditions for stability are derived for the presented setup in terms of the update time h and the parameters of the plant and of its model. The deterioration of behavior when either h or the modeling error increase is explicitly shown.

2. INTRODUCTION

2.1 BACKGROUND

The use of networks as a media to interconnect the different components in an industrial control system is rapidly increasing. For example in large scale plants and in geographically distributed systems, where the number and/or location of different subsystems to control make the use of single wires to interconnect the control system prohibitively expensive. In addition, the flexibility and ease of maintenance of a system using a network to transfer information is a very appealing goal. Systems designed in this manner allow for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when component failure occurs, and in general they allow having a high level supervisor control over the entire plant.

There are many examples in which placing a network to interconnect control applications is convenient. A typical example is the case of aircraft control. In this case, different sensors and control surfaces among other control components are distributed over the aircraft. Another example is the case of manufacturing factories where it is a common practice to implement data acquisition systems along the process path. Dozens of sensors are deployed over critical points to make important information about the process available to quality control engineers. Most of the times, these sensors will transmit the collected information to a central computer using an industrial network. More than often the need to create new control loops appears as quality or industrial engineers analyze the data retrieved by the acquisition network. In this case, it seems natural to attach the controllers and actuators to the already existent network and share the data already provided by the deployed sensors. In general, the use of a network on a control system is desirable when there is a large number of distributed sensors and actuators.

One of the main problems to be addressed when considering a networked control system is the large amount of bandwidth required by each subsystem. In traditional control the feedback path makes the sensor information available to the controller continuously. Industrial controllers usually connect the sensor to the controller using a wire. Bandwidth and dynamic response of a plant are closely related. The faster the dynamics of the plant, the larger is its bandwidth. This usually translates to large frequency content on the controlling signal and a continuous exchange of information between the plant and the controller.

In this report, we consider the problem of having a sensor that is connected to the actuator/controller by a network; that is, the feedback path is a network. However, data networks typically have limited bandwidth and transfer information in a discrete time framework, and this makes the task of designing a control system rather challenging.

To overcome the bandwidth constraint several approaches have been proposed. In [1] Brockett introduces the notion of minimum attention control that attempts to reduce the time and state feedback dependence of the control law. This can be viewed as a tradeoff between open loop and closed loop control. Since the network is placed on the feedback path, open loop control uses no feedback bandwidth but has poor performance in reducing steady state error, rejecting noise and disturbances, and reducing the sensitivity of the system to parameter variations. The shortcomings of open loop control under uncertainties are well known. On the other hand closed loop control has an excellent performance but typically requires significant bandwidth. In [1] the problem is posed as an optimization problem. An attention functional to be minimized is defined penalizing $\|\partial u / \partial t\|$ and $\|\partial u / \partial x\|$ to obtain the control law u that has the least dependence on time t and the plant state x. It is claimed that the derived control law requires less frequent updating and is more robust to small changes in the data. Intuitively, one can think "large values of $\|\partial u / \partial x\|$ indicate closed loop control while large values of $\|\partial u / \partial t\|$ indicate open loop control." So by varying the ratio of the penalties in the attention functional over the two terms one can arrive to a tradeoff between the computational-intensive, frequently updated, and bandwidth hungry closed loop control and the easy-to-implement open loop control strategies.

In [7] Nair et al. study the case of an infinite dimensional time varying discrete plant with unknown initial condition. The plant is being controlled using a network on the feedback path. The only constraint on the network is that a finite set of symbols can be used to send the information from the sensor to the controller/actuator. The sensor then implements a coder that transmits the information to the controller/actuator at each sampling time. The information takes negligible time to get to its destination and the data corruption probability is assumed to be zero. The result is that, under certain technical conditions on the probability density function of the initial condition, the plant is stabilizable asymptotically in the m-th output moment and in the infinite horizon if and only if the coder and controller comply with certain characteristics that depend on the alphabet size and some dynamical constants. The special case in which the plant is unstable and LTI the condition is reduced to have $R > \log_2 |\lambda|$ where R is the transmission rate in bits per second and λ is the unstable open loop pole with largest magnitude.

Elia et al. [2] propose the design of a quantized controller and state estimator for a LTI discrete system. The result is an optimized controller and state estimator that operates in discrete periodic times and quantized values for the state. It is obvious that the coarser the discretization, the less the bandwidth required for the system to work. This work follows the same line as the one by Nair et al. It is shown that the coarsest or least dense quantizer that quadratically stabilizes the plant is logarithmic and can be computed by solving a special LQR problem. The theory is then extended to continuous LTI plants using constant sampling time intervals. It is shown that the optimal sampling interval time (using the proposed quantizers) is only a function of the sum of the unstable eigenvalues of the system. Francis et al further explores the idea to prevent chattering in the system in [11].

Together with the optimization of the sampling period and quantizer some effort has been done in optimizing the sampling times and control law. This can be view as type of scheduling. Several approaches have been proposed. Rehbinder et al [10] proves the intuitive idea that the plant with fastest dynamics should be given more network bandwidth resources. In the same spirit, Xiao et al study the optimization of the word length, the output scaling, and the controller or estimator gain in [22]. Various communication schemes are presented and analyzed.

The optimization of switching times and state estimation through a network is covered by A.S. Matveev et al in [20]. In this paper it is studied a linear discrete-time partially observed system perturbed by a white noise. The observations are transmitted to the controller via communication channels with irregular transmission times. Various measurements signals may incur independent delays or arrive at the estimator out of order. The estimator can dynamically control which sensors will it connect to. The minimum variance state estimate and the optimal sensor switching strategy are obtained. Basically a Kalman-like state estimator that is able to connect to its inputs a limited array of the plant output sensors and deal with a variable delay to optimally estimate the plant state.

Bauer et al. analyze the problem on a network with random delays in [3]. The paper proposes the use of a Smith predictor in a discrete framework to eliminate the delay induced by the network. The Smith predictor is placed in front of the controller and uses knowledge about the plant to propagate forward the delayed information from the sensor and make it accessible to the controller. We will use the intuitive idea that knowledge about the plant dynamics can help to relax the network quality of service requirements without sacrificing the performance of the networked control system.

In [4], Walsh presents a protocol that uses dynamic scheduling and a zero order hold at the controller input. The notion of maximal allowable transfer interval, MATI, is introduced to place an upper bound on the time between transfers of information from the sensor to the controller. In this case the controller is designed without taking the network into account, a desirable feature. However, serious behavior degradation can result if the MATI is too large and the network slow. Also a dynamic scheduling is introduced: Try-Once-Discard or TOD protocol. In TOD each sensor has a transmission priority that is proportional to the error between the last data sent and the actual measured value. The sensor with biggest error is given maximum priority to transmit. Additionally, if a sensor is denied access to the network by contention, it will discard the packet and construct a new one with fresh data before trying again to transmit. These results are extended to non-linear plants in [16]. Tolerance of this systems under different types of noise is studied on [17].

The effects of different scheduling schemes for the TOD protocol are studied in [18]. It is implied that the plant performance is improved if an appropriate scheduling scheme is used. Scheduling is of utmost importance when there is a number of sensors, actuators and controllers competing for network resources. It determines the nature of the delays, transmission rates, etc. A deterministic scheduling scheme is presented by Hristu-Varsakelis in [24]. Deterministic communication sequences are easier to analyze and sometimes can have a superior performance than non-deterministic scheduling schemes but can also be difficult to enforce.

In [14] Beldiman, Walsh and Bushnell extend the results in [4] to include a state predictor, for LTI systems, to estimate the state in between updates. Two types of state predictors are defined. The first one is the so-called open loop predictor. Which is basically a plant model that is updated with an invertible transformation of the state vector available at the plant output. The model assumes complete knowledge of the plant. In other words, there is no uncertainty in the model dynamics and they match perfectly with the plant dynamics. Once the model's state has been updated, it can provide the controller with an estimate of the plant state vector. The second predictor, called closed loop predictor, has the same structure as a Lurenberg observer. It receives the output of the plant and tries to recreate the plant output in between transmissions. This predicted output is then fed to the controller. This closed observer needs the network to be very fast in order for the observer to converge. Sufficient conditions are given for the stability of this NCS setup. Our work further extends the idea to use an approximate model of the plant. The main difference between the results of [14] and the ones presented here are that we don't assume to have complete knowledge of the plant. Thus, some amount of uncertainty is allowed in the plant model. The special structure of this NCS allows the characterization of necessary and sufficient conditions that turn to be less conservative and easier to check than those from [14].

In [15] Ishii and Francis, extend their Dwell Time Controller [11] for systems with output feedback. In the dwell time controller setup the plant's output is fed to a state observer. The estimated state is then quantized using a logarithmic partition of the state space. The quantized value is sent through the network to the controller/actuator. After decoding the message the controller will apply a constant input to the plant that corresponds to the received value of the quantized state. The logarithmic partition is coarser as the state's distance to the origin is bigger, and it is finer when the state is closer to the origin. This seems to be reasonable since fine steering of the state is more useful when the state is close to the origin. The logarithmic partitions are made overlapping so that the system can tolerate some noise generated by the sensors. Also a dwell time is specified to reduce fast chattering produced by the controller when switching control inputs. To do so, the time interval between switchings is enforced to be bigger or equal than the dwell time. This can be done by timing the messages sent to the controller. In our approach we will use the natural choice of having a state observer at the sensor side of the network.

In [19] synthesis and existence of a networked optimal controller for nonstationary linear parameter-varying (LPV) systems are shown. The plant has a spatially distributed structure and thus is natural to think in a networked controller structure. The paper presents sufficient convex conditions for the existence and construction of an "admissible" controller. This controller is based on the plant structure, that is, the controller has the same distributed LPV structure as the plant. This allows for a number of efficient computational schemes for controller implementation, as well as a direct method of truncating the controller to obtain a decentralized distributed controller. We recognize the importance of including the plant structure in the controller to facilitate the construction of an efficient controller.

In [20] it is presented an algorithm for the stabilization of a multi-input/ multi-output discrete time linear system via a limited capacity channel. The approach taken is a deterministic multi-rate state space approach that leads to a nonlinear dynamic feedback controller. The network channel is assumed to be noiseless and with time delays associated with transmissions. An important feature of the approach is that it is a multi-rate approach in which symbols are transmitted across the channel at a slower rate that the control inputs are applied to the discrete time plant. It is also shown how the results can be extended to the general output feedback case by using a form of deadbeat observer. Both coder and decoder are dynamic systems that are "synchronized" by an evolving state that is known at both sides. The state-space is partitioned dynamically as the system approaches its steady state, in this way asymptotic behavior is proved achievable. The actuator (or sensor) can use the state synchronization proposed here to predict the behavior of the sensor (or actuator) in the intervals where there is no communication. This idea is exploited here by having an evolving state that is "synchronized" on both sides of the communication channel.

In [23] A.S. Matveev presents a NCS with an estimator/central controller, and several semi-independent subsystems. The central controller receives information from the different subsystems about the uncontrollable dynamics. It compresses and processes the data and sends it to the different subsystems. The data arrives to the local controllers at each subsystem. The local controller selects the right information from the central controller message. This message contains propagated versions of the estimated state and a time stamp so that the local controller can choose the right propagated version. This is done since the central controller does not know the value of the transmission delay. The local controller estimates the control-induced part of the controllable states of the sub system and computes the state of the controllable state by adding the term corresponding to the uncontrollable dynamics that was received from the central controller. The problem is solved in a quadratic optimization framework.

2.2. PROBLEM SETUP AND RATIONALE

It is clear that the reduction of bandwidth necessitated by the communication network in a networked control system is a major concern. This can perhaps be addressed by two methods: the first being minimizing the transfer of information between the sensor and the controller/actuator. The second method is to compress or reduce the size of the data transferred at each transaction.

Actually deployed and popular networks in the industry include CAN bus, PROFIBUS, Fieldbus Foundation, and Ethernet among others. Each of these protocols and standards has very different characteristics such as network contention resolution or scheduling schemes, transmission media, etc. Among the shared characteristics are the small transport time and big overhead (network control information included in the packet). This means that data compression by reducing the size of the data transmitted has negligible effects over the overall system performance. *So reducing the number of packets transmitted brings better benefits than data compression*. The reduction of the number of packets transmitted through the network can translate into larger minimal transfer times between the components. It is also to be noted that any delay in an information transaction is usually due to network access contention. This translates into what has been already noted by Walsh in [8]: that the sensor with a fast sampling rate can send through the network the latest data available resulting in a negligible information transfer delay. But there will still be contention in the network so that, *even though the delay is small, the sensor data would not be available at all times to the controller/actuator*. This brings us again to the idea of reducing the data transfer rate as much as possible. In this manner more bandwidth will be available to allocate more resources without sacrificing stability and ultimately performance of the overall system.

We will consider the case where the controller and the actuator are combined together into a single node. That is, the network is between the sensor and the controller/actuator node. Assuming that the controller and the actuator physically coexist is reasonable since embedded microprocessors are usually incorporated into the actuator to process the data received by the network and execute the commands received.

In this report we will concentrate on characterizing the transfer time between the sensor and the actuator. The transfer time is the time between information exchanges from the sensor to the controller/actuator. Thus, the inverse of this transfer time would be the frequency at which the sensor will send information to the actuator. Our goal will be to identify the maximum transfer time between the sensor and the actuator while keeping the system stable. This will reduce the bandwidth required from the network and will free it for other tasks such as other control loops using the network and/or non-control information exchange. In order to increase the transfer time we will use the knowledge we have of the plant dynamics. The plant model is used at the controller/actuator side to recreate the plant behavior so that the sensor can delay sending data since the model can provide an approximation of the plant dynamics. *The main idea is to perform the feedback by updating the model's state using the actual state of the plant that is provided by the sensor. The rest of the time the control action is based on a plant model that is incorporated in the controller/actuator and is running open loop for a period of h seconds.* The setup is pictured below.

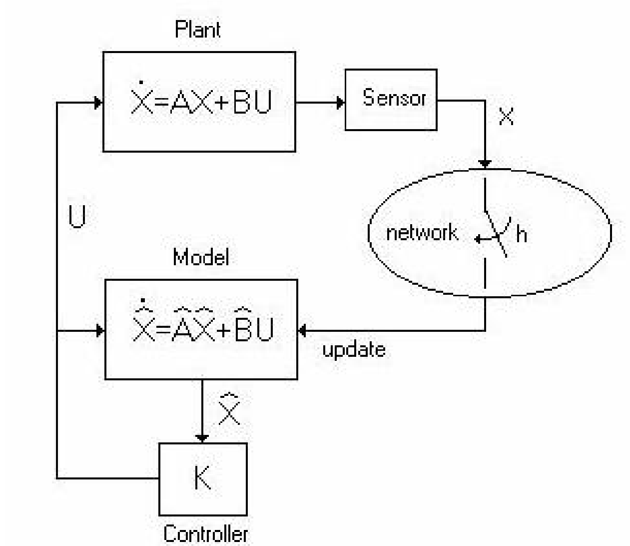


Figure 1.

This idea of a tradeoff between open loop and closed loop control is related to the minimal attention control proposed by Brockett in [1]. One of the main differences resides in that minimal attention control makes this tradeoff in a continuous way. The resulting controller works in a similarly to a sampled data system. The lack of awareness of the controller of the intersampling plant behavior usually results in performance degradation. In our setup the tradeoff between open loop and closed loop control is done in a “discrete” manner. Having knowledge of the plant at the actuator side enables us to run the plant in open loop, while the update of the model state provides the close loop information needed to overcome model uncertainties and plant disturbances.

Walsh uses a similar setup to that in Figure 1 in [4] except that the controller is a dynamic system that receives the output of the plant as a direct input to it. The controller can be implemented such that it includes knowledge of the plant by the setup but it doesn’t implement a state update in the same manner as the setup presented here. The input to the controller is maintained the same way as a zero order hold does. If the controller were designed without taking into account the network it would not achieve the same performance which it was designed for. More over, the results presented in [4] are only sufficient and conservative as it is shown latter on this report.

Our approach is novel in that it incorporates a model of the plant the state of which is updated with the plant’s state. We also present a necessary and sufficient condition for its stability that results in a maximum transfer time that depends solely on the model inaccuracies. In the absence of plant disturbances arbitrarily long transfer times can be achieved depending on modeling errors.

This report is organized as follows: in section 3 necessary and sufficient conditions are developed for the setup showed in Figure 1. This is the case where the state vector is directly measurable and sent through the network to the controller/actuator. It is shown that the networked control system depicted in Figure 1 is globally exponentially stable if and only if the eigenvalues of a test matrix M are inside the unit circle. This matrix M depends on the plant dynamics, the model uncertainties and the model update time. A numeric example with simulations is given in section 4. Section 5 further analyzes the structure of this matrix. The special case of a first order plant is given in section 5.1 with an example and simulations in section 5.2. The case where the plant state vector is not directly measurable is considered in section 6. Here a state estimator is used at the output of the plant to estimate the plant state vector so that it can be used at the other end of the network to update the model at the controller/actuator. Necessary and sufficient conditions are given for this extended setup. The condition is given over a new matrix with a similar structure to the one used for the state feedback case. Section 6.1 illustrates the previous results with an example and simulations. In section 6.2 the structure of the test matrix is analyzed for the case of output feedback. All the results are then extended to the case of discrete plants in section 7, with the case of state feedback in section 7.1 and output feedback in section 7.2. An example for the case of state feedback with discrete plants is given in section 7.3. Section 8 considers the case of a full state feedback networked system with constant delays. It is shown that a necessary and sufficient condition to obtain stability is that the eigenvalues of a test matrix, similar to the previous ones, must be inside the unit circle. Conclusions are presented in section 9. Appendix A describes the procedure to obtain the Maximum Allowable Transfer Interval according to [4] that is used to compare results in section 4. Appendices B through H present the proofs for theorems 3 through 9.

3. FULL STATE FEEDBACK CONTROL

If all the states are available, then the sensors can send this information through the network to update the model vector state. For our analysis we will assume that the compensated model is stable and that the transportation delay is negligible, which is, completely justifiable in most of the popular network standards like CAN bus or Ethernet. We will assume that the frequency at which the network updates the state in the controller is constant. This last assumption can be relaxed for certain cases. The idea is to find the smallest frequency at which the network must update the state in the controller, that is, an upper bound for *h*, the update time. Note that, in our approach, we don't make any assumptions on the plant dynamics. Usual assumptions include requiring a stable plant or a smaller update than the sampling time in the case of a discrete controller. Here we assume that the plant may be unstable.

We will now characterize the control system depicted in Figure 1. Were the system dynamics are given by:

$$\text{Plant}: \quad \dot{x} = Ax + Bu$$

$$\text{Model}: \quad \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u$$

$$\text{Controller}: \quad u = K\hat{x}$$

Since the sensor has the full state vector available, its function will be to send the state information through the network every h seconds. Now we will introduce some definitions. The state error is defined as $e = x - \hat{x}$, and represents the difference between the plant state and the model state. The modeling error matrices $\tilde{A} = A - \hat{A}$, $\tilde{B} = B - \hat{B}$, represent the difference between the plant and the model. Finally, the update times are t_k, where $t_k - t_{k-1} = h$ for all k. Since the model state is updated every t_k seconds, $e(t_k) = 0$ for $k = 0,1,2,\ldots$. This resetting of the state error every update time is a key factor in our control system.

Now for $t \in [t_k, t_{k+1})$, we have that:

$$u = K\hat{x}$$

so

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

with initial conditions $\hat{x}(t_k) = x(t_k)$

Introducing the error $e(t) = x(t) - \hat{x}(t)$, it is very easy to see that the dynamics of the overall system for $t \in [t_k, t_{k+1})$ can be described by

$$\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \textit{ and } \begin{bmatrix} x(t_k) \\ e(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ 0 \end{bmatrix}, \qquad \forall t \in [t_k, t_{k+1}), \textit{ with } t_{k+1} - t_k = h \tag{1}$$

Define $z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}$, and $\Lambda = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix}$ so that (1) can be rewritten as $\dot{z} = \Lambda z$ for $t \in [t_k, t_{k+1})$.

We will now derive the response of the networked system. Then we will show under what conditions the system will be stable.

THEOREM #1

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

$$z(t) = e^{\Lambda(t - t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0$$

$$t \in [t_k, t_{k+1}), \textit{ with } t_{k+1} - t_k = h$$

Proof.

On the interval $t \in [t_k, t_{k+1})$, the system response is

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix} = e^{\Lambda(t-t_k)} z(t_k) \tag{2}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix}$, that is, the error *e(t)* is reset to zero. We can represent this by

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z(t_k^-)$$

Using (2) to calculate $z(t_k^-)$ we obtain

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1})$$

In view of (2) we have that if at time $t=t_0$, $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned} z(t) &= e^{\Lambda(t-t_k)} z(t_k) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-2}) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-3}) \\ &\ldots \\ &= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k z_0 \end{aligned} \tag{3}$$

Now we know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h}$ is of the form $\begin{bmatrix} M & N \\ 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k$ has the form $\begin{bmatrix} M^K & P \\ 0 & 0 \end{bmatrix}$.

Additionally we note the special form of the initial condition $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ so that

$$\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M^k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \tag{4}$$

In view (4) it is clear that we can represent the system response as:

$$z(t) = e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0$$
$$t \in [t_k, t_{k+1}),\ with\ \ t_{k+1} - t_k = h \tag{5}$$

♦

A necessary and sufficient condition for stability of the networked system will now be derived. For this the following definition for global exponential stability [5] is needed.

DEFINITION #1

The equilibrium $z = 0$ *of a system described by* $\dot{z} = f(t, z)$ *with initial condition* $z(t_0) = z_0$ *is exponentially stable at large (or globally) if there exists* $\alpha > 0$ *and for any* $\beta > 0$, *there exists* $k(\beta) > 0$ *such that the solution*

$$\left\| \phi(t, t_0, z_0) \right\| \le k(\beta) \left\| z_0 \right\| e^{-\alpha(t-t_0)}, \quad \forall t \ge t_0$$

whenever $\left\| z_0 \right\| < \beta$.

With this definition of stability we state the following theorem characterizing the necessary and sufficient conditions for the system described by (1) to have global exponential stability around the solution $z = 0$. The norm used here is the 2-norm but any other consistent norm can also be used.

THEOREM #2

The system described by (1) is globally exponentially stable around the solution $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle.*

Proof.

Sufficiency.

Taking the norm of the solution described as in Theorem #1:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \le \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\| \tag{7}$$

Now lets analyze the first term on the right hand side of (7):

$$\left\| e^{\Lambda(t-t_k)} \right\| \le 1 + (t-t_k)\overline{\sigma}(\Lambda) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots = e^{\overline{\sigma}(\Lambda)(t-t_k)} \le e^{\overline{\sigma}(\Lambda)h} = K_1 \tag{8}$$

where $\overline{\sigma}(\Lambda)$ is the largest singular value of Λ. In general this term can always be bounded since the time difference $t-t_k$ is always smaller than *h*. In other words even when Λ has eigenvalues with positive real part, $\left\| e^{\Lambda(t-t_k)} \right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I\,0 \\ 0\,0 \end{bmatrix} \right)^k \right\| \le K_2 e^{-\alpha_1 k} \tag{9}$$

with $K_2, \alpha_1 > 0$.
Since *k* is a function of time we can bound the right term of (9) in terms of *t*:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h}t} = K_3 e^{-\alpha t} \tag{10}$$

with $K_3, \alpha > 0$.
So from (7) using (8) and (10) we can conclude:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \le K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\| \tag{11}$$

Necessity.

We will now prove the necessity part of the theorem by contradiction. Assume the system described by (1) is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times t_{k+1}^{-}, that is, just before the update. We will concentrate on a specific term: the state of the plant $x(t_{k+1}^{-})$, which is the first element of $z(t_{k+1}^{-})$. We will call $x(t_{k+1}^{-})$, $\xi(k)$.

Now assume $e^{\Lambda\tau}$ has the following form:

$$e^{\Lambda\tau} = \begin{bmatrix} W(\tau) & X(\tau) \\ Y(\tau) & Z(\tau) \end{bmatrix}$$

then we can express the solution $z(t)$ as:

$$\begin{aligned} & e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \\ &= \begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(t-t_k)(W(h))^k & 0 \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0 \end{aligned} \tag{12}$$

Now the values of the solution at times t_{k+1}^{-}, that is just before the update, are

$$z(t_{k+1}^{-}) = \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 \tag{13}$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first element of $z(t_{k+1}^{-})$, which we call $\xi(k)$, will in general grow with *k*. In other words we can't ensure $\xi(k)$ will converge to zero for general initial condition x_0.

$$\left\| x(t_{k+1}^{-}) \right\| = \left\| \xi(k) \right\| = \left\| (W(h))^{k+1} x_0 \right\| \to \infty \quad as \quad k \to \infty$$

this clearly means the system cannot be stable, and thus we have a contradiction.

♦

4. EXAMPLE OF FULL STATE FEEDBACK

Consider the following unstable plant:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We will use the state feedback controller given by $u = Kx$ with $K = [-1 \quad -2]$.

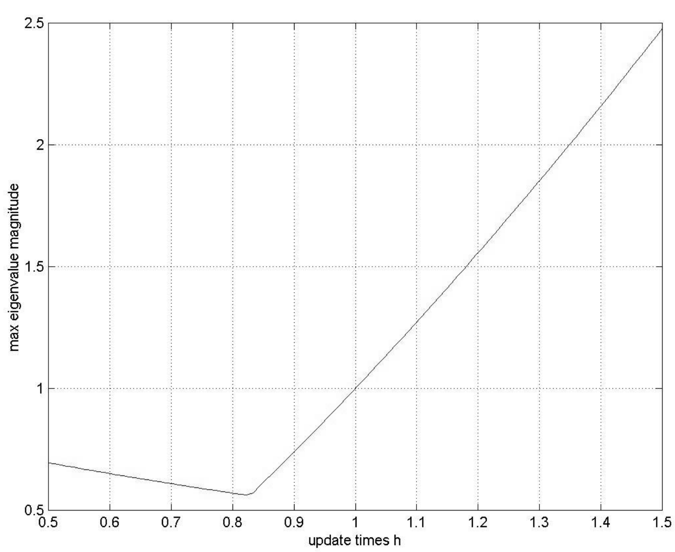
Usually it is assumed that the actuator/controller will hold the last value received from the sensor until the next time the sensor transmits. Under this assumption the controller/actuator node acts as if having a zero order hold at its input. We will first analyze this situation. To do so, we will transform the plant model so that it holds the last state update presented to it by the network. In this manner the controller will behave in the exact same manner as if having a zero order hold at its input of the state vector update. The model this way modified would be:

$$\hat{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \hat{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

So now we need to search for the biggest h such that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has its eigenvalues inside the unit circle. In this case Λ is given by:

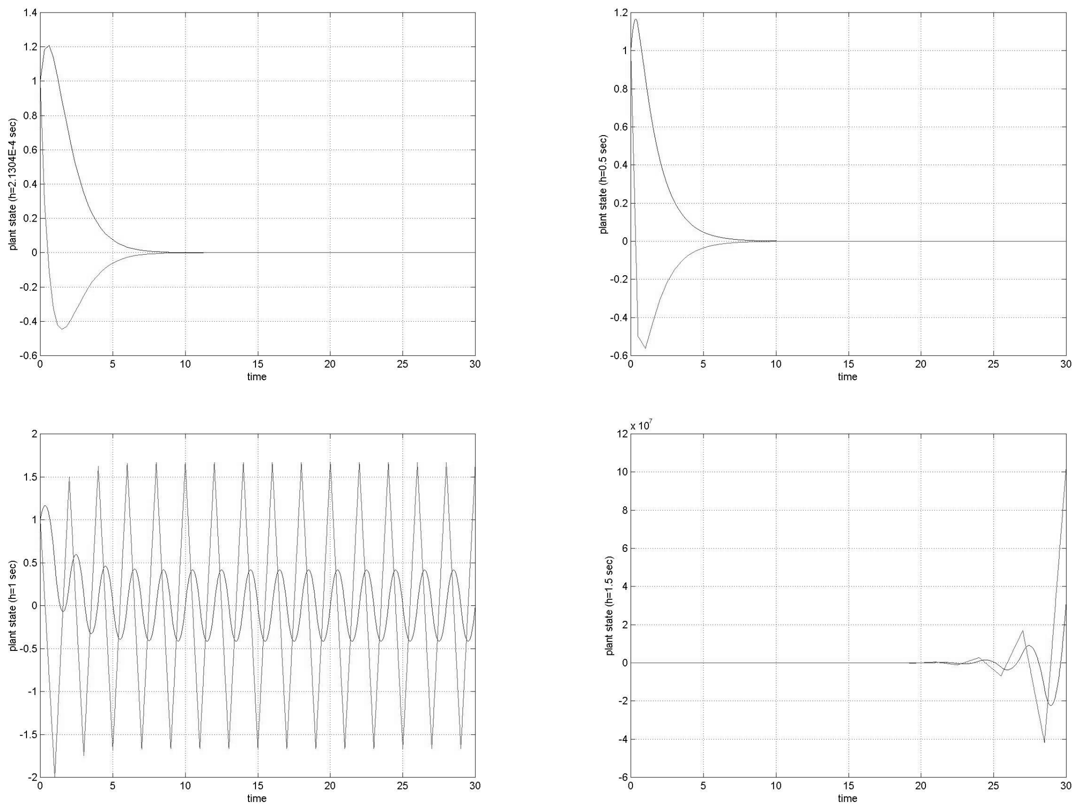
$$\Lambda = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -2 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ -1 & -2 & 1 & 2 \end{bmatrix}$$

To do so we plotted the maximum eigenvalue magnitude versus the update time:



From the picture we see that the condition for stability is to have $h < 1$ second. In fact the test matrix M will have one eigenvalue with magnitude of 1 for h=1 second. If we use the results by [4] we would have obtained that in order to stabilize the system we would need to have h<2.1304E-4, which is very conservative. A derivation for the bounds of h using [4] is given in Appendix A.

Simulations of the system with update times of 2.1304E-4, 0.5, 1 and 1.5 seconds are shown below. Note that the plant was initialized with an initial condition of $[1\ 1]^T$.

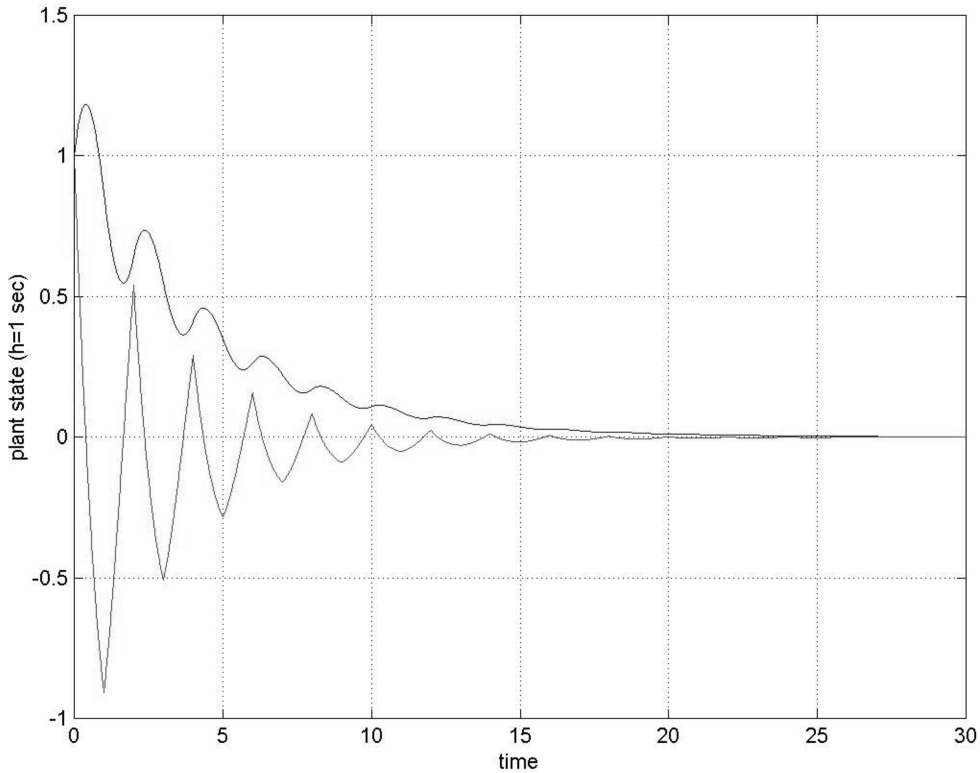


It can be seen that for h=1 second the system is marginally stable, and for h=1.5 seconds is completely unstable. It is also clear that the performance obtained with h=0.5 seconds is not too different to the one obtained with h=2.1304E-4 seconds, but the difference in the amount of bandwidth used is large. If we were to use Ethernet that has a minimum message size of 72bytes (including preamble bits and start of delimiter fields) the data rate would be 2.7Mbits/sec for the case of h=2.1304E-4 seconds, and 1.2Kbits/sec for the case of h=0.5 seconds.

Now using our setup, we will use a plant model that has a similar structure to the actual plant. We will use the randomly perturbed plant model:

$$\hat{A} = \begin{bmatrix} -0.5395 & 1.7990 \\ -0.7126 & -0.4972 \end{bmatrix},\ \hat{B} = \begin{bmatrix} 0.3030 \\ 0.0096 \end{bmatrix}$$

This plant model gives a test matrix M maximum eigenvalue magnitude of 0.7986 for an update time of h=1 which was our actual boundary for our previous example. The system response is pictured below.



Comparing this response with the one obtained with an update time of 0.5 seconds, we only see a slight degradation of performance: the settling time has been increased from 7 seconds to approximately 16 seconds. This has been obtained even though the plant model barely resembles the original plant and that the update time has been increased in 50%.

5. CHARACTERIZATION OF THE EIGENVALUES IN THE TEST MATRIX M

It is of interest to study the eigenvalues of the networked control system matrix $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ and express them, if possible, in terms of h and the error in the plant model $\tilde{A}$ and $\tilde{B}$. To do so, we first apply a transformation to the matrix Λ to obtain a diagonal matrix that will facilitate the computation of the exponential part.

We choose the transformation $P = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$ with inverse $P^{-1} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$. Applying this transformation over Λ we obtain:

$$\overline{\Lambda} = P\Lambda P^{-1} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} A+BK & -BK \\ \tilde{A}+\tilde{B}K & \hat{A}-\tilde{B}K \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A}+\hat{B}K \end{bmatrix}$$

Using this transformation we obtain:

$$M=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} P^{-1}e^{\overline{\Lambda}h}P\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\overline{\Lambda}h}\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}$$

The matrix exponential $e^{\overline{\Lambda}h}$ may be found directly or by considering a Laplace transform based approach. For the latter approach we will change the variable h to t.

$$L\{M\}=L\left\{\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\overline{\Lambda}t}\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}\right\}=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} L\left\{e^{\overline{\Lambda}t}\right\}\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}$$

$$=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} (sI-A)^{-1} & (sI-A)^{-1}BK(sI-(\hat{A}+\hat{B}K))^{-1} \\ 0 & (sI-(\hat{A}+\hat{B}K))^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}$$

$$=\begin{bmatrix} (sI-A)^{-1}+(sI-A)^{-1}BK(sI-(\hat{A}+\hat{B}K))^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Note that only the upper left block contains the critical eigenvalues. Using inverse Laplace transform:

$$\begin{aligned}
&L^{-1}\{(sI-A)^{-1}+(sI-A)^{-1}BK(sI-(\hat{A}+\hat{B}K))^{-1}\} \\
&=L^{-1}\{(sI-A)^{-1}(I+BK(sI-\hat{A}-\hat{B}K)^{-1})\} \\
&=L^{-1}\{(sI-A)^{-1}(sI-\hat{A}-\hat{B}K+BK)(sI-\hat{A}-\hat{B}K)^{-1}\} \\
&=L^{-1}\{(sI-A)^{-1}(sI-A+\tilde{A}+\tilde{B}K)(sI-\hat{A}-\hat{B}K)^{-1}\} \\
&=L^{-1}\{(I+(sI-A)^{-1}(\tilde{A}+\tilde{B}K))(sI-\hat{A}-\hat{B}K)^{-1}\} \\
&=L^{-1}\{(sI-\hat{A}-\hat{B}K)^{-1}+(sI-A)^{-1}(\tilde{A}+\tilde{B}K)(sI-\hat{A}-\hat{B}K)^{-1}\} \\
&=e^{(\hat{A}+\hat{B}K)t}+e^{At}\int_0^t e^{-A\tau}(\tilde{A}+\tilde{B}K)e^{(\hat{A}+\hat{B}K)\tau}d\tau
\end{aligned}$$

That is the eigenvalues in question are exactly the eigenvalues of:

$$e^{(\hat{A}+\hat{B}K)h}+e^{Ah}\int_0^h e^{-A\tau}(\tilde{A}+\tilde{B}K)e^{(\hat{A}+\hat{B}K)\tau}d\tau$$

In view of Theorem #2 the following lemma has been proved:

LEMMA #1

The system described by (1) is globally exponentially stable around the solution $z=\begin{bmatrix} x \\ e \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *iff the eigenvalues of* $I=e^{(\hat{A}+\hat{B}K)h}+e^{Ah}\int_0^h e^{-A\tau}(\tilde{A}+\tilde{B}K)e^{(\hat{A}+\hat{B}K)\tau}d\tau$ *are strictly inside the unit circle.*

One can gain a better insight of the system by observing the structure of I. To start with, we observe that the eigenvalues of the compensated model appear in the first term of I. In that sense we can see the term $D=e^{Ah}\int_0^h e^{-A\tau}(\tilde{A}+\tilde{B}K)e^{(\hat{A}+\hat{B}K)\tau}d\tau$ as a perturbation over the desired eigenvalues. Even if the eigenvalues of the original plant were unstable the perturbation D can be made small enough by having h and $\tilde{A}+\tilde{B}K$ small and thus minimize their impact over the eigenvalues of the compensated plant. We also observe that if the update time h is driven to zero, then D=0. Also it is possible to make D=0 by having a model that is exact. This agrees with the intuition that if the model has exactly the same dynamics as the plant then the system will have the desired behavior regardless of how long is the update time h.

5.1 FIRST-ORDER PLANTS

If the plant is of first order then the integral expression in I can be evaluated. Furthermore, I will have the value of the eigenvalue to be checked. Evaluating I we obtain:

$$
\begin{aligned}
I &= e^{(\hat{A}+\hat{B}K)h} + e^{Ah}\int_0^h e^{-A\tau}(\tilde{A}+\tilde{B}K)e^{(\hat{A}+\hat{B}K)\tau}d\tau \\
&= e^{(\hat{A}+\hat{B}K)h} + e^{Ah}(\tilde{A}+\tilde{B}K)\int_0^h e^{-A\tau}e^{(\hat{A}+\hat{B}K)\tau}d\tau \\
&= e^{(\hat{A}+\hat{B}K)h} + e^{Ah}(\tilde{A}+\tilde{B}K)(e^{-Ah}e^{(\hat{A}+\hat{B}K)h}-1)/(\hat{A}+\hat{B}K-A) \\
&= e^{(\hat{A}+\hat{B}K)h} + (\tilde{A}+\tilde{B}K)(e^{(\hat{A}+\hat{B}K)h}-e^{Ah})/(\hat{A}+\hat{B}K-A) \\
&= e^{(\hat{A}+\hat{B}K)h}(1+(\tilde{A}+\tilde{B}K)/(\hat{A}+\hat{B}K-A)) - e^{Ah}(\tilde{A}+\tilde{B}K)/(\hat{A}+\hat{B}K-A) \\
&= e^{(\hat{A}+\hat{B}K)h}(1+(\tilde{A}+\tilde{B}K)/(\hat{A}+\hat{B}K-A)) - e^{Ah}(\tilde{A}+\tilde{B}K)/(\hat{A}+\hat{B}K-A) \\
&= e^{Ah}(1+BK/(\tilde{A}+\tilde{B}K-BK)) - e^{(\hat{A}+\hat{B}K)h}(BK/(\tilde{A}+\tilde{B}K-BK)) \\
&= e^{Ah}F(BK,\tilde{A}+\tilde{B}K) - e^{(\hat{A}+\hat{B}K)h}G(BK,\tilde{A}+\tilde{B}K)
\end{aligned}
$$

We can think of the functions F and G as weighting functions. Note that they depend on the variable $\tilde{A}+\tilde{B}K$. We observe that if $\tilde{A}+\tilde{B}K=0$ then $F(BK,0)=0$ and $G(BK,0)=-1$. This will result in $I=e^{(\hat{A}+\hat{B}K)h}$ which reflect the eigenvalues of the model. We also note that in general if $\tilde{A}+\tilde{B}K \ll BK$ then $F(BK,\tilde{A}+\tilde{B}K)\cong 0$ and $G(BK,\tilde{A}+\tilde{B}K)\cong -1$.

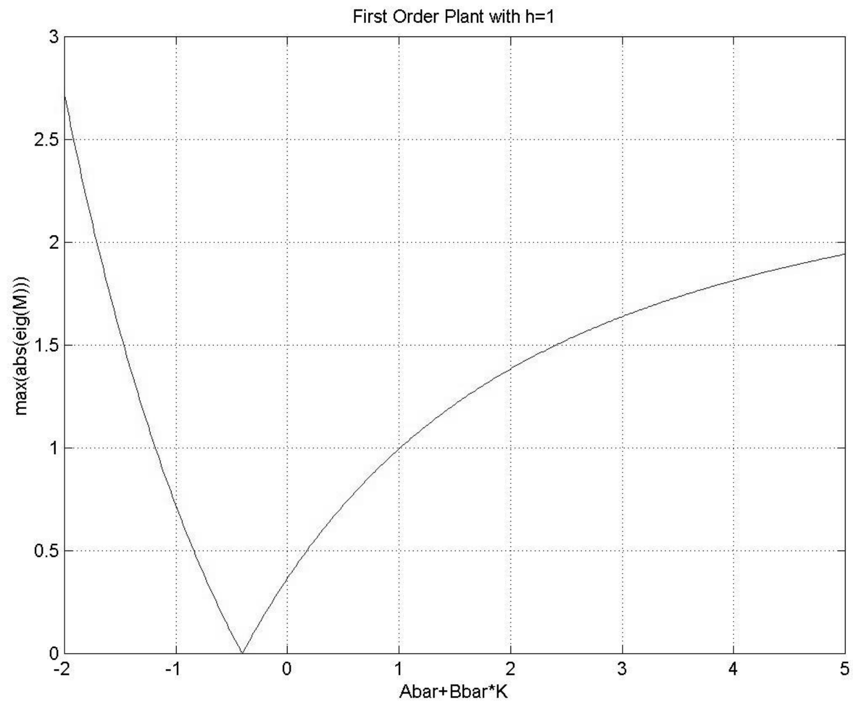
The last thing to ensure for the eigenvalues of the original uncompensated plant to vanish is that the update time h is small enough so that it doesn't surpass the attenuation provided by F. *In general, h should be chosen to be such that* e^{Ah} *is small*.

The fact that F and G depend on $\tilde{A}+\tilde{B}K$ simplifies the analysis of I. Given a h, I can be plotted as a function of only $\tilde{A}+\tilde{B}K$. Surfaces can be generated if h is also considered.

5.2 EXAMPLE OF A FIRST ORDER PLANT

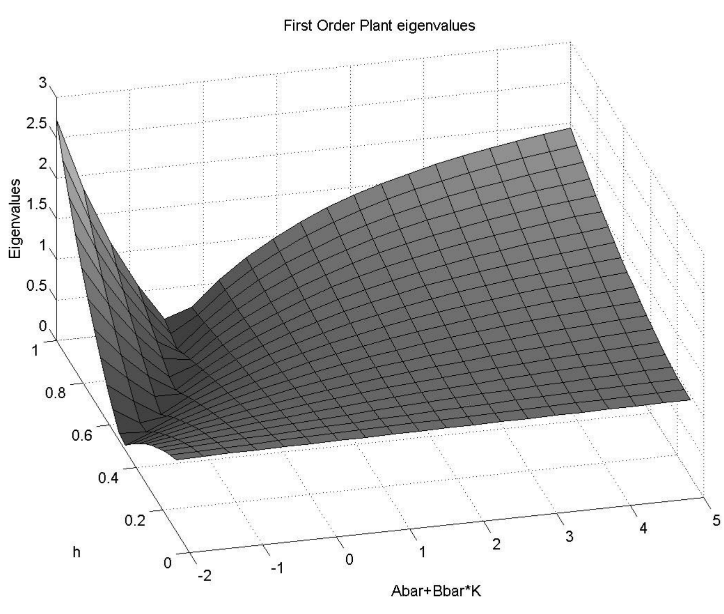
We will now present an example for the behavior of the magnitude of the largest eigenvalue of the test matrix. Consider the first order plant: $\dot{x}=x+u$. Thus the state space description is $A=1,\ B=1$. This a clearly unstable plant that we will stabilize using a state gain K=-2.

We now plot the magnitude of the largest eigenvalue using an update time h=1 as a function of the difference $\tilde{A}+\tilde{B}K$ between the compensated plant and the compensated model.



We see that the system will be stable if $\tilde{A}+\tilde{B}K$ is approximately between -1.25 and 1. We also see that zero magnitude eigenvalues are achievable.

We now plot the surface corresponding to the magnitude of the eigenvalues of the test matrix as a function of $\tilde{A}+\tilde{B}K$ and the update time h.



6. OUTPUT FEEDBACK PLANTS

We have been considering only plants where the full state vector is available at the output. We now extend our approach to include plants where the state is not directly measurable. In this case, in order to obtain an estimate of the plant state vector, a state observer is used. It is assumed that the state observer is collocated with the sensor. Again, we use the plant model, $\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u$, to design the state observer. See Figure 2.

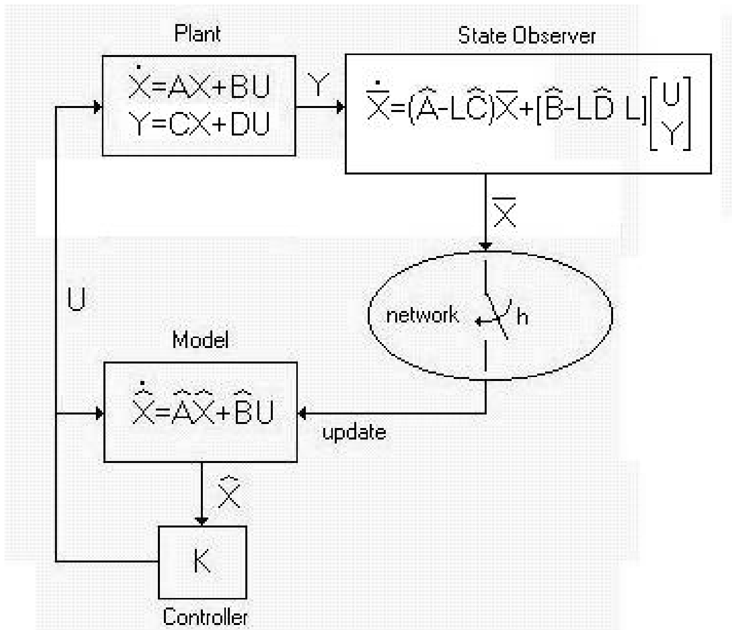


Figure 2.

Having the sensor carry the computational load of an observer is justified by the fact that typically sensors that can be connected to a network have an embedded processor inside. This processor is usually in charge of performing the sampling, filtering and implementing the network layer services required to connect to the network. Ishii and Francis give a similar justification in [15]. In their approach an observer is placed at the output of the plant to reconstruct the state vector. The result is then quantized and sent over the network to the controller.

The observer has as inputs the output and input of the plant. To acquire the input of the plant, which is at the other side of the communication link, the observer can have a version of the model and controller. In this way, the output of the controller, that is the input of the plant, can be simultaneously and continuously generated at both ends of the feedback path with the only requirement that the observer makes sure that model has been updated. This last requirement ensures that both the controller and the observer are synchronized. Handshaking provided by most network protocols can be used.

The observer has the form of a Lurenberg observer with gain L. It makes use of the plant model:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u$$
$$y = \hat{C}\hat{x} + \hat{D}u$$

In summary, the system dynamic equations are:

$$\begin{aligned}
&\text{Plant}:\\
&\quad \dot{x} = Ax + Bu\\
&\quad y = Cx + Du\\[1em]
&\text{Model}:\\
&\quad \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u\\
&\quad y = \hat{C}\hat{x} + \hat{D}u\\[1em]
&\text{Controller}:\\
&\quad u = K\hat{x}\\[1em]
&\text{Observer}:\\
&\quad \dot{\bar{x}} = (\hat{A} - L\hat{C})\bar{x} + \begin{bmatrix}\hat{B} - L\hat{D} & L\end{bmatrix}\begin{bmatrix}u\\ y\end{bmatrix}\\
&\quad \text{for } t \in [t_k, t_{k+1})
\end{aligned} \tag{14}$$

We now proceed in a similar way as in the previous case of full feedback. Namely, there will be an update interval *h*, after which the observer updates the controller's model state $\hat{x}$ with its estimate $\bar{x}$. We will also define an error *e* that will be the difference between the controller's model state and the observer's estimate: $e = \bar{x} - \hat{x}$.

It is clear that at times t_k, where $t_k - t_{k-1} = h$, the error *e* will be equal to zero.

$$e(t) = \begin{Bmatrix} \bar{x}(t) - \hat{x}(t) & t \in (t_k, t_{k+1}) \\ 0 & t = t_k \end{Bmatrix} \tag{15}$$

Also we will define the modeling error matrices in the same way as before:

$$\tilde{A} = A - \hat{A}$$
$$\tilde{B} = B - \hat{B}$$
$$\tilde{C} = C - \hat{C}$$
$$\tilde{D} = D - \hat{D}$$

Now for $t \in [t_k, t_{k+1})$, we have that:

$$u = K\hat{x}$$

So we have:

$$\begin{aligned}
\dot{x} &= Ax + BK\hat{x} \\
\dot{\hat{x}} &= (\hat{A} + \hat{B}K)\hat{x} \\
\dot{\bar{x}} &= (\hat{A} - L\hat{C})\bar{x} + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} K\hat{x} \\ Cx + DK\hat{x} \end{bmatrix} \\
&= \begin{bmatrix} LC & \hat{B}K + L\tilde{D}K & \hat{A} - L\hat{C} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \\ \bar{x} \end{bmatrix}
\end{aligned}$$

with initial conditions $\hat{x}(t_k) = \bar{x}(t_k)$. Using the same approach as before, we express the system dynamics in terms of the states that will not change on the update times.

Then the dynamics of the overall system for $t \in [t_k, t_{k+1})$ can be described by

$$\begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \\ e \end{bmatrix}$$
$$t \in [t_k, t_{k+1}), \text{ with } t_{k+1} - t_k = h$$
$$\text{and} \begin{bmatrix} x(t_k) \\ \bar{x}(t_k) \\ e(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ \bar{x}(t_k^-) \\ 0 \end{bmatrix}. \tag{16}$$

Define $z = \begin{bmatrix} x \\ \bar{x} \\ e \end{bmatrix}$, and $\Lambda = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix}$ so that (16) can be represented by $\dot{z} = \Lambda z$ for $t \in [t_k, t_{k+1})$.

THEOREM #3

The system with dynamics described by (16) with initial conditions $z(t_0) = \begin{bmatrix} x(t_0) \\ \bar{x}(t_0) \\ 0 \end{bmatrix} = z_0$, $t_0 = 0$, *, has the following response:*

$$z(t) = e^{\Lambda(t - t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0$$
$$t \in [t_k, t_{k+1}), \textit{ with } t_{k+1} - t_k = h$$

The proof for this theorem is analogous to that of Theorem #1. See Appendix B. We will present now the necessary and sufficient conditions for this system to be exponentially stable at large (or globally).

THEOREM #4

The system described by (16) is globally exponentially stable around the solution $z = \begin{bmatrix} x \\ \bar{x} \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ *are inside the unit circle.*

Again the proof for this theorem is almost identical to the one for theorem #2. Even easier to visualize the similarity is to group the plant state and the state estimator state into one single state $\breve{x} = \begin{bmatrix} x \\ \bar{x} \end{bmatrix}$ and use the same techniques used for theorems #1 and #2. The complete proof is presented in Appendix C.

6.1 EXAMPLE OF OUTPUT FEEDBACK SYSTEM

We now present an example using a double integrator as the plant. This is the same plant used in the full state feedback example. The plant dynamics are given by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \end{bmatrix}; D = 0$$

We will use the state feedback controller $K = \begin{bmatrix} -1 & -2 \end{bmatrix}$. This controller if used with the plant will place its eigenvalues at –1. Now a state estimator with gain $L = \begin{bmatrix} 20 \\ 100 \end{bmatrix}$ is used to place the state observer eigenvalues at –10. The model used is a perturbation of the original plant:

$$\hat{A} = \begin{bmatrix} 0.0958 & 1.0604 \\ -0.0066 & -0.0134 \end{bmatrix}; \hat{B} = \begin{bmatrix} -0.0518 \\ 1.0269 \end{bmatrix}; \hat{C} = \begin{bmatrix} 0.9734 & -0.0137 \end{bmatrix}; \hat{D} = -0.0396$$

We now plot the magnitude of the largest eigenvalue for the test matrix versus the update time:

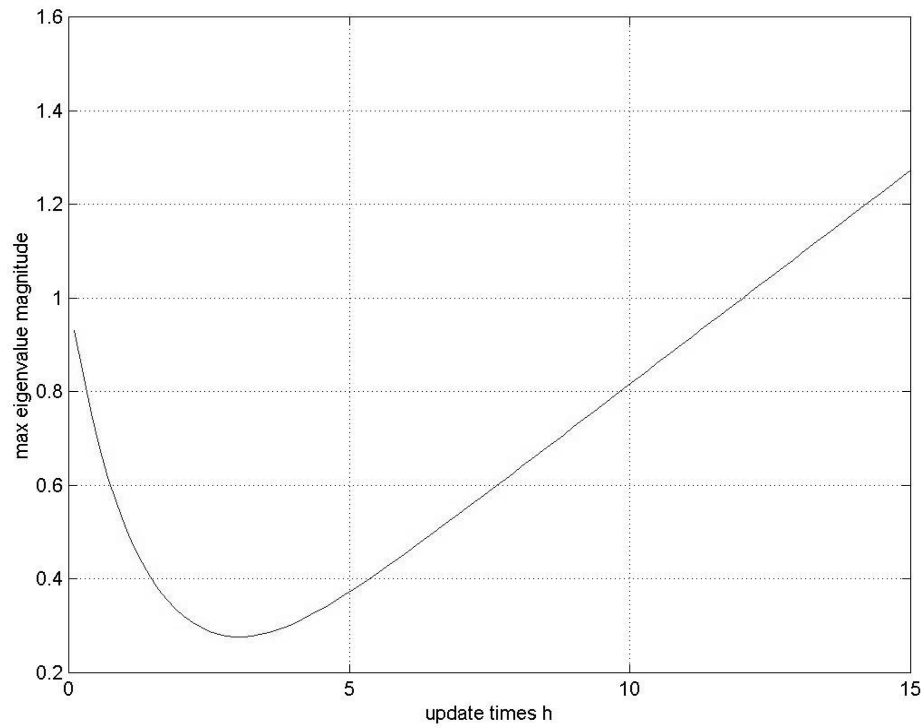


Figure 3.

Stability is observed for an update time up to 12 seconds.

A simulation of the system with an update time of 1 second, an initial state of the plant at [1 1]', and zero initial conditions for the estimator and controller's state is shown below.

Luis A. Montestruque and Panos J. Antsaklis, "Model-Based Networked Control Systems - Stability," ISIS Technical Report ISIS-2002-001, January 2002.

The updates are noticeable in the plant model state at times 1, 2, and 3 seconds.

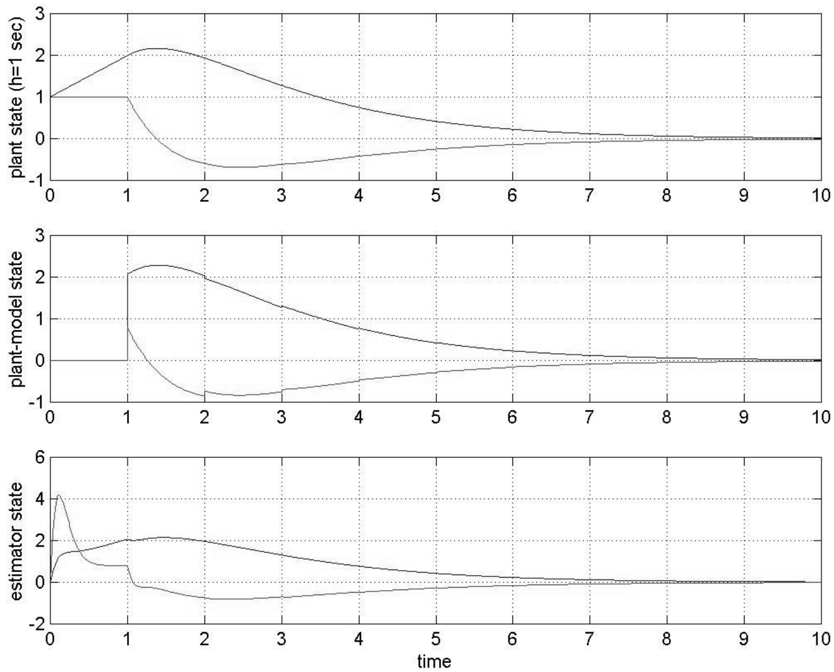


Figure 4.

6.2. CHARACTERIZATION OF THE EIGENVALUES OF THE TEST MATRIX *M*

Here we characterize the nature of the test matrix for the case of output feedback networked control. We will do this is a similar fashion as for the full state feedback networked control. The system dynamics are given by:

$$\dot{x} = Ax + BK\hat{x}$$

$$\dot{\hat{x}} = (\hat{A} + \hat{B}K)\hat{x}$$

$$\dot{\bar{x}} = \begin{bmatrix} LC & \hat{B}K + L\tilde{D}K & \hat{A} - L\hat{C} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \\ \bar{x} \end{bmatrix}$$

A common assumption is that the D matrix is zero, we will further assume that $\tilde{D} = 0$. We can now arrange the states in a vector to get a compact representation:

$$\begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 & BK \\ LC & \hat{A} - L\hat{C} & \hat{B}K \\ 0 & 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \\ \hat{x} \end{bmatrix} = \overline{\Lambda} \begin{bmatrix} x \\ \bar{x} \\ \hat{x} \end{bmatrix}$$

This representation has the advantage to have a block triangular structure. Additionally, we note that

$$\Lambda = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & -I \end{bmatrix} \overline{\Lambda} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & -I \end{bmatrix}$$

The test matrix M can now be represented in terms of $\overline{\Lambda}$:

$$\begin{aligned} M &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & -I \end{bmatrix} e^{\overline{\Lambda}h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & -I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\overline{\Lambda}h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & 0 \end{bmatrix} \end{aligned}$$

We will now replace the h by t and use the Laplace transform to facilitate the manipulation of the test matrix.

$$L\{e^{\overline{\Lambda}t}\} =$$
$$= \begin{bmatrix} (sI-A)^{-1} & 0 & (sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} \\ \left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1} & \left(sI-\hat{A}+L\hat{C}\right)^{-1} & \begin{array}{c}\left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} \\ +\left(sI-\hat{A}+L\hat{C}\right)^{-1}\hat{B}K\left(sI-\hat{A}-\hat{B}K\right)^{-1}\end{array} \\ 0 & 0 & \left(sI-\hat{A}-\hat{B}K\right)^{-1} \end{bmatrix}$$

Now we can compute the Laplace transform of the test matrix:

$$L\{M\} =$$
$$= \begin{bmatrix} (sI-A)^{-1} & (sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} & 0 \\ \left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1} & \begin{array}{c}\left(sI-\hat{A}+L\hat{C}\right)^{-1}+\left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} \\ +\left(sI-\hat{A}+L\hat{C}\right)^{-1}\hat{B}K\left(sI-\hat{A}-\hat{B}K\right)^{-1}\end{array} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We can now extract the part of the test matrix M that could have eigenvalues inside the unit circle. We will call this new test matrix N.

$$N =$$
$$= L^{-1}\left\{\begin{bmatrix} (sI-A)^{-1} & (sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} \\ \left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1} & \begin{array}{c}\left(sI-\hat{A}+L\hat{C}\right)^{-1}+\left(sI-\hat{A}+L\hat{C}\right)^{-1}LC(sI-A)^{-1}BK\left(sI-\hat{A}-\hat{B}K\right)^{-1} \\ +\left(sI-\hat{A}+L\hat{C}\right)^{-1}\hat{B}K\left(sI-\hat{A}-\hat{B}K\right)^{-1}\end{array} \end{bmatrix}\right\}$$

We will apply a linear transformation to the matrix N that will let us examine its structure closer.

$$N' = \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} N \begin{bmatrix} I & 0 \\ I & I \end{bmatrix}$$

Doing this and some further manipulation we obtain

$$N' = L^{-1}\left\{\begin{bmatrix} \left(sI - \hat{A} - \hat{B}K\right)^{-1} + \varepsilon_1 & (sI - A)^{-1} BK\left(sI - \hat{A} - \hat{B}K\right)^{-1} \\ \varepsilon_3 & \left(sI - \hat{A} + L\hat{C}\right)^{-1} + \varepsilon_2 \end{bmatrix}\right\}$$

with

$$\varepsilon_1 = (sI - A)^{-1}(\tilde{A} + \tilde{B}K)(sI - \hat{A} - \hat{B}K)^{-1}$$
$$\varepsilon_2 = (sI - \hat{A} + L\hat{C})^{-1}\left(\tilde{B}K - \left(\tilde{A} - L\tilde{C}\right)(sI - A)^{-1} BK\right)(sI - \hat{A} - \hat{B}K)^{-1}$$
$$\varepsilon_3 = (sI - \hat{A} + L\hat{C})^{-1}\left(\tilde{A} - L\tilde{C} + \tilde{B}K - \left(\tilde{A} - L\tilde{C}\right)(sI - A)^{-1}\left(\tilde{A} + \tilde{B}K\right)\right)(sI - \hat{A} - \hat{B}K)^{-1}$$

It is clear that if $\tilde{A} \to 0$, $\tilde{B} \to 0$, and $\tilde{C} \to 0$ then $\varepsilon_1 \to 0$, $\varepsilon_2 \to 0$, and $\varepsilon_3 \to 0$. These matrices ε_1, ε_2, and ε_3 can be considered as disturbances over the standard controller-observer characteristic matrix $\begin{bmatrix} \left(sI - \hat{A} - \hat{B}K\right)^{-1} & (sI - A)^{-1} BK\left(sI - \hat{A} - \hat{B}K\right)^{-1} \\ 0 & \left(sI - \hat{A} + L\hat{C}\right)^{-1} \end{bmatrix}$.

7. THE DISCRETE TIME DOMAIN

So far we have applied our results to continuous plants. We will extend our results to discrete time plants of the form:

$$x(n+1) = Ax(n) + Bu(n)$$
$$y(n) = Cx(n) + Du(n)$$

There are some few assumptions we need to make before we carry our results over to the discrete time domain. In order to have appropriate updates from the sensor side to the actuator side we must ensure that both are synchronized in the sense that both will carry out their respective tasks at the same time tic. Moreover updates will be carried over at some of those time instants. *In this case that the update interval h will be an integer number, representing after how many clock tics will the actuator's model be updated. This means that the sensor only will need to send the state or output information once every h samples.*

We will now present the corresponding theorems for the discrete systems with state observer and without state observer.

Lets consider the following equations:

$$
\begin{aligned}
&\text{Plant}: \\
&\qquad x(n+1) = Ax(n) + Bu(n) \\
&\qquad y(n) = Cx(n) + Du(n) \\
&\text{Model}: \\
&\qquad \hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n) \\
&\qquad y(n) = \hat{C}\hat{x}(n) + \hat{D}u(n) \\
&\text{Controller}: \\
&\qquad u(n) = K\hat{x}(n) \\
&\text{Observer}: \\
&\qquad \bar{x}(n+1) = (\hat{A} - L\hat{C})\bar{x}(n) + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} u(n) \\ y(n) \end{bmatrix} \\
&\text{for } n \in [n_k, n_{k+1}), \text{ with } n_{k+1} - n_k = h
\end{aligned}
\tag{17}
$$

The procedure is quite similar to the one used for the continuous case. We must only note that the update interval h is an integer. We now present the results for the full state feedback case.

7.1 FULL STATE FEEDBACK

The approach is exactly the same that the one used for continuous plants. The results carry over with a slight difference in the test matrix. Is in this difference where the discrete nature of the plant is made evident. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described by

$$
\begin{aligned}
&\begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} \\
&n \in [n_k, n_{k+1}),\ n_k - n_{k+1} = h \\
&\text{and } e(n) = 0.
\end{aligned}
\tag{18}
$$

Define $z = \begin{bmatrix} x \\ e \end{bmatrix}$, and $\Lambda = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix}$ so that (18) can be represented by $z(n+1) = \Lambda z(n)$ for $n \in [n_k, n_{k+1})$.

THEOREM #5

The system described by (18) with initial conditions $z(n_0)=\begin{bmatrix} x(n_0) \\ 0 \end{bmatrix}=z_0$, *has the following response:*

$$z(n)=\Lambda^{n-n_k}\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k z_0$$
$$n\in[n_k,n_{k+1}),n_k-n_{k+1}=h$$

We note that the only difference between this theorem and the continuous version is in the transition matrix used for the dynamics of the system in between updates. For the proof see Appendix D.

We now introduce an exponential global stability definition for the case of discrete plants. This definition is very similar to the one for the continuous case.

DEFINITION #2

The equilibrium $z=0$ *of a system described by* $\dot{z}=f(t,z)$ *with initial condition* $z(n_0)=z_0$ *is exponentially stable at large (or globally) if there exists* $\alpha>0$ *and for any* $\beta>0$, *there exists* $k(\beta)>0$ *such that the solution*

$$\left\|\phi(n,n_0,z_0)\right\|\le k(\beta)\left\|z_0\right\|e^{-\alpha(n-n_0)},\quad \forall n\ge n_0$$

whenever $\left\|z_0\right\|<\beta$.

With this definition of stability we scan now state the corresponding necessary and sufficient conditions for the exponential global stability of the system described by (18). Again the norm used here is the 2-norm but any other consistent norm can also be used.

THEOREM #6

The system described by (18) is globally exponentially stable around the solution $z=\begin{bmatrix} x \\ e \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are inside the unit circle.*

The proof for this theorem is shown in Appendix E. We now present the equivalent theorems for the case with a state observer at the sensor side.

7.2 OUTPUT FEEDBACK

It not always possible to measure directly the plant state vector. So, as with the continuous plant case, we extend the previous result to include a Lurenberg state observer at the output of the plant. This observer will sample, estimate and send the state estimate every h samples. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described by

$$\begin{bmatrix} x(n+1) \\ \bar{x}(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix} \begin{bmatrix} x(n) \\ \bar{x}(n) \\ e(n) \end{bmatrix}$$
$$n \in [n_k, n_{k+1}), \text{ with } n_{k+1} - n_k = h$$
$$\text{and } e(n_k) = 0. \tag{19}$$

Define $z = \begin{bmatrix} x \\ \bar{x} \\ e \end{bmatrix}$, and $\Lambda = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix}$ so that (19) can be represented by $z(n+1) = \Lambda z(n)$ for $n \in [n_k, n_{k+1})$.

THEOREM #7

The system with dynamics described by (19) with initial conditions $z(n_0) = \begin{bmatrix} x(n_0) \\ \bar{x}(n_0) \\ 0 \end{bmatrix} = z_0$, *has the following response:*

$$z(n) = \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0$$
$$n \in [n_k, n_{k+1}), \textit{ with } n_{k+1} - n_k = h$$

Proof provided in Appendix F.

THEOREM #8

The system described by (19) is globally exponentially stable around the solution $z = \begin{bmatrix} x \\ \bar{x} \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ *are inside the unit circle.*

The proof for Theorem #8 is provided in Appendix G.

7.3 EXAMPLE OF FULL STATE FEEDBACK ON A DISCRETE PLANT

We will now present an example of the full state feedback setup using a double integrator plant of the form:

$$x(n+1)=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}x(n)+\begin{bmatrix}0\\ 1\end{bmatrix}u(n)$$

With state feedback law $u(n)=\begin{bmatrix}-1 & -2\end{bmatrix}x(n)$. In this case the feedback corresponds to a deadbeat controller, in other words the non-networked plant will have its eigenvalues at {0,0}.

We generated a plant model using a random perturbation of the original plant matrices:

$$\hat{x}(n+1)=\begin{bmatrix}1.3626 & 1.6636\\ -0.2410 & 1.0056\end{bmatrix}\hat{x}(n)+\begin{bmatrix}0.4189\\ 0.8578\end{bmatrix}u(n)$$

Below is the plot the magnitude of the largest eigenvalue of the test matrix $\begin{bmatrix}I & 0\\ 0 & 0\end{bmatrix}\Lambda^h\begin{bmatrix}I & 0\\ 0 & 0\end{bmatrix}$. We must note that the minimum value h can take is one. This would correspond to the case where the network delivers instantly the state of the plant at each clock tic. *With h=1 the networked control system will behave exactly as the non-networked control system since every the model will follow the plant dynamics. This is because the sensor will update the model at every sampling time.* It would be the equivalent of having h=0 for the continuous plant case.

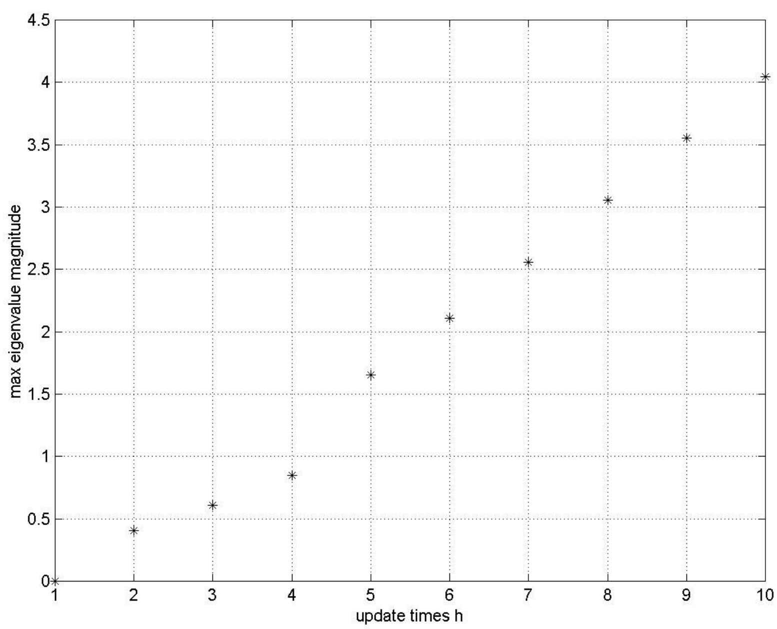


Figure 5.

From the graph it can be seen that the maximum value for h is 4. For $h \geq 5$ the NCS has eigenvalues with magnitude larger than one and therefore will be unstable.

Now we show a plot of the response of the system with different values of update times h. We selected the values of h=3 samples and 5 samples to show the case of a stable and an unstable NCS.

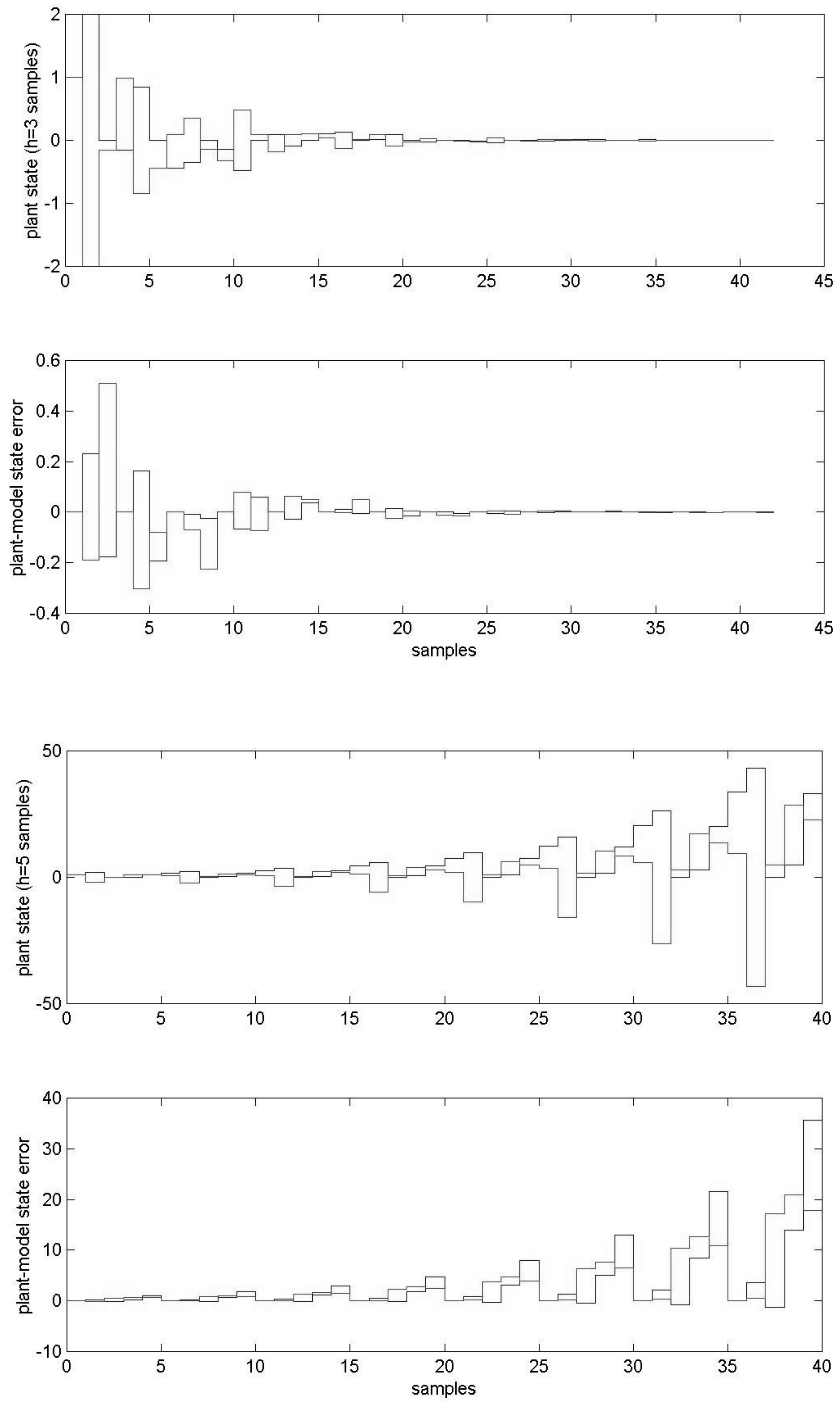


Figure 6.

8. NETWORK DELAYS

Previously we assumed that the network delays were negligible. This is usually true for plants with slow dynamics and networks with relatively big bandwidth. When this is not the case the network delay cannot be neglected. Network delays can occur for many reasons. There are three important delay sources:

- Processing time.
- Media access contention.
- Propagation and transmission time.

The first one, processing time, occurs on both ends of the communication channel. On the transmitter, the processing time is the time elapsed between the transmitting process makes the request to the operating system to transmit a message, and when the message is ready to be sent. And in the receiver this is the time interval that occurs between the last bit of the message is received by the receiver, and when the message is delivered by the operative system to the receiver process.

The media access contention time is the time the transmitter has to wait until the communication channel is not busy. This usually the case when several transmitters have to share the same media.

The propagation and transmission time is the time the message takes to be placed on the network media and to travel through the network to reach the receiver. In local area networks the time the message takes to travel or propagate through the media is small in comparison to wide area networks or internetworks like the Internet. The time the message takes to be placed on the network depends on the size of the message and the baud rate.

If the control network is a local area network, as is common practice in industry, the propagation and transmission time can be established forehand with good accuracy. A similar thing occurs with the processing time. If real time operating systems are used the processing time can be accurately calculated. Finally media access contention delay can be fixed with the use of a communication protocol with scheduling. Fast data communication networks like Token Ring, Token Bus, and ArcNet fall into this category. Industry oriented control networks like Foundation Fieldbus also implements a scheduler through its LAS or Link Active Scheduler. Even the inherently non-deterministic Ethernet has addressed the problem of not having a specified contention time with the so-called Switched Ethernet.

In conclusion most of these delays can be at least bounded if the network conditions are appropriate.

Next we extend our results to include the case were transmission delay is present. We will assume that the update time h is larger than the delay time τ. As before we will assume that the update time h is constant. We will also assume at this time that the delay τ is constant. We will present here the case of full state feedback systems.

So, at times $kh-\tau$ the sensor transmits the state data to the controller/actuator. This data will arrive τ seconds latter. So, at times kh the controller/actuator receives the state vector value $x(kh-\tau)$. The main idea is to use the plant model in the controller/actuator to calculate the present value of the state. After this, the state approximate obtained can be used to update the controller's model as in previous setups. The system is depicted below.

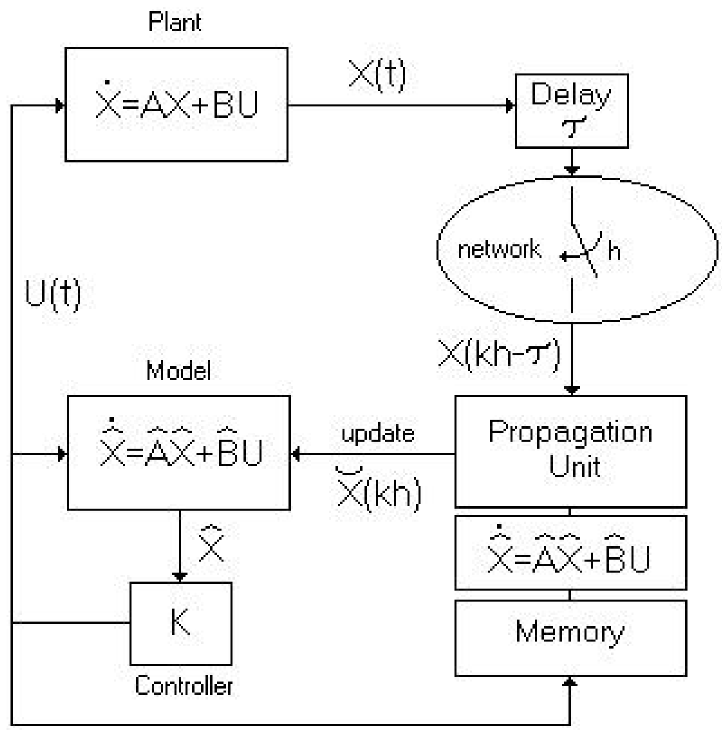


Figure 7.

The Propagation Unit uses the plant model and the past values of the control input $u(t)$ to calculate an estimate of actual state $\breve{x}(kh)$ from the received data $x(kh-\tau)$. This estimate is then used to update the model that with the controller will generate the control signal for the plant.

The system is then described by the following equations:

$$\begin{aligned}
&\text{Plant}: \\
&\qquad \dot{x} = Ax + Bu \\
&\text{Model}: \\
&\qquad \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \\
&\text{Controller}: \\
&\qquad u = K\hat{x} \\
&\qquad\qquad\qquad t \in [t_k, t_{k+1}) \\
&\text{Propagation Unit}: \\
&\qquad \dot{\breve{x}} = \hat{A}\breve{x} + \hat{B}u \\
&\qquad\qquad\qquad t \in [t_{k+1} - \tau, t_{k+1}] \\
&\text{Update law}: \\
&\qquad \text{at } t = t_{k+1} - \tau : \breve{x} \leftarrow x \\
&\qquad \text{at } t = t_{k+1} \quad : \hat{x} \leftarrow \breve{x}
\end{aligned} \tag{20}$$

To ease the analysis, we initialize the propagation unit at time t_{k+1}-τ with the state vector that the sensor obtains. We then run the plant, model, and propagation unit together until t_{k+1}. At this time, the model is updated with the propagation unit state vector, as described in the update law of (20). This is equivalent to have the propagation unit receive the state vector $x(t_{k+1}-\tau)$ at t_{k+1} and propagate it instantaneously to t_{k+1}.

We define the errors $\hat{e} = \breve{x} - \hat{x}$ and $\breve{e} = x - \breve{x}$. We also make the following definitions:

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

$$z = \begin{bmatrix} x \\ \breve{e} \\ \hat{e} \end{bmatrix}, \quad \tilde{A} = A - \hat{A}, \quad \tilde{B} = B - \hat{B}.$$

With these definitions we proceed to present the system described by (20) in a compact form. The dynamics of the overall system for $t \in [t_k, t_{k+1})$ can be described by

$$\dot{z}(t) = \Lambda z(t),\; z(t_k) = \begin{bmatrix} x(t_k^-) \\ \breve{e}(t_k^-) \\ 0 \end{bmatrix}, \qquad t \in [t_k, t_{k+1} - \tau),$$

$$\dot{z}(t) = \Lambda z(t),\; z(t_{k+1} - \tau) = \begin{bmatrix} x((t_{k+1} - \tau)^-) \\ 0 \\ \breve{e}((t_{k+1} - \tau)^-) + \hat{e}((t_{k+1} - \tau)^-) \end{bmatrix}, \qquad t \in [t_{k+1} - \tau, t_{k+1}),$$

with $t_{k+1} - t_k = h,\; 0 < \tau < h.$

(21)

THEOREM #9

The system described by (21) is globally exponentially stable around the solution $z = \begin{bmatrix} x \\ \breve{e} \\ \hat{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}$ *are inside the unit circle.*

The proof for Theorem #9 is provided in Appendix H.

9. CONCLUSIONS

The presented setup represents a natural way of placing critical information about the plant on the network so to reduce the data traffic load. By making the sensor and actuator more "intelligent" the NCS is able to predict the future behavior of the plant, and send the precise information at critical times so to ensure the plant stability. The presence of computational load at any end of the feedback path is not considered a limitation of the applicability of the presented setups given the advances in microcomputing. Most of the sensors and actuators available in the market have a microcontroller embedded that is in charge of a number of tasks. For our case it is clear that at least they should implement network services. So it seams reasonable to have them perform these computationally inexpensive operations like a state observer or a state feedback simulation.

Easy to verify and/or enforce conditions are derived here as a result of the simplicity of the setup. The systems placed at the sensor and actuator/controller can be seen as having the effect of reducing the sampling rate for maintaining the system stable. This is very clear in the case of the discrete plant where the controller only needs to receive 1 sample every h samples the sensor has available to transmit.

Another extension would be the one in which the update time can vary with time. Time varying matrix stability tests can be performed over the test matrix. But, given the special structure of these matrices, simple and direct conditions should be obtained.

Performance is of main concern also. The techniques used in this report are very similar to the ones known as lifting operators [9, 12, 13]. It can be shown that plant induced norms are invariant under these lifting operators. The resulting system is very similar to a digital plant and therefore allows the use of well known techniques to ensure system performance. H_∞ and H_2 control optimization can be used to obtain optimal controllers and observer gains.

APPENDIX A

The plant is given by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The state feedback controller given by $u = Kx$ with $K = \begin{bmatrix} -1 & -2 \end{bmatrix}$.

We will obtain now the MATI (Maximum Allowable Transfer Interval) according to the result in [4] by Walsh et al. To do so we will make the state of the plant available to be transmitted over the network and the dynamic controller to behave as a state feedback controller with gain *K*. The plant will then be given by:

$$A_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B_p = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with dynamics given by:

$$\dot{x}_p = A_p x_p + B_p u,\ y = C_p x_p$$

The controller is described by:

$$A_c = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix},\ B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ C_c = \begin{bmatrix} 0 & 0 \end{bmatrix},\ D_c = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

with dynamics described by:

$$\dot{x}_c = A_c x_c + B_c \hat{y},\ u = C_p x_p - D_p \hat{y}$$

where $\hat{y}$ is the latest information of the plant acquired by controller through the network.

For the case where there is a single sensor node operating, the result in [4] claims that if the MATI *h* satisfies:

$$h < \min\left\{ \frac{\ln(2)}{\|A\|}, \frac{1}{8\|A\|\left(\sqrt{\lambda_2/\lambda_1}+1\right)}, \frac{1}{16\|A\|^2 \lambda_2 \sqrt{\lambda_2/\lambda_1}\left(\sqrt{\lambda_2/\lambda_1}+1\right)} \right\}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$A_{11} = \begin{bmatrix} A_p - B_p D_c C_p & B_p C_c \\ -B_c C_p & A_c \end{bmatrix},$$

$$A_{12} = \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix},$$

$$A_{21} = \begin{bmatrix} C_p A_p - C_p B_p D_c C_p & C_p B_p C_c \end{bmatrix},$$

$$A_{22} = C_p B_p D_c,$$

$$\lambda_1 = \lambda_{\min}(P), \lambda_2 = \lambda_{\max}(P),\ A_{11}^T P + P A_{11} = -I$$

then the networked control system is globally asymptotically stable.
For the plant previously presented this results in:

$$h < 2.1304\text{E-}4 \text{ seconds.}$$

♦

APPENDIX B

Proof of Theorem #3.

On the interval $t \in [t_k, t_{k+1})$, the system response is

$$z(t) = \begin{bmatrix} x(t) \\ \bar{x}(t) \\ e(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \begin{bmatrix} x(t_k) \\ \bar{x}(t_k) \\ 0 \end{bmatrix} = e^{\Lambda(t-t_k)} z(t_k) \tag{B2}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ \bar{x}(t_k) \\ 0 \end{bmatrix}$, that is, the error *e(t)* is reset to zero. We can represent this by

$$z(t_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} z(t_k^-)$$

Using (B2) to calculate $z(t_k^-)$ we obtain

$$z(t_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1})$$

In view of (B2) we have that if at time $t=t_0$, $z(t_0)=z_0=\begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned}
z(t) &= e^{\Lambda(t-t_k)} z(t_k) \\
&= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}) \\
&= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-2}) \\
&= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-3}) \\
&\dots \\
&= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k z_0
\end{aligned}$$

(B3)

Now we know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}$ is of the form $\begin{bmatrix} M_1 & M_2 & N_1 \\ M_3 & M_4 & N_2 \\ 0 & 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k$ has the form $\begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \\ [0 \quad 0] & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(t_0)=z_0=\begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix}$ so that

$$\begin{aligned}
&\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k \begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k \begin{bmatrix} x_0 \\ \bar{x}_0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ [0 \quad 0] & 0 \end{bmatrix} \\
&= \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ [0 \quad 0] & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix} = \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix}
\end{aligned}$$

(B4)

In view (B4) it is clear that we can represent the system response as:

$$z(t) = e^{\Lambda(t-t_k)}\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k z_0$$

$$t \in [t_k, t_{k+1}),\ with \quad t_{k+1} - t_k = h$$

♦

APPENDIX C

Proof of Theorem #4.

Sufficiency.

Taking the norm of the solution described as in Theorem #3:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)}\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k z_0 \right\| \le \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k \right\| \cdot \|z_0\| \tag{C7}$$

Now lets analyze the first term on the right hand side of (C7):

$$\left\| e^{\Lambda(t-t_k)} \right\| \le 1 + (t-t_k)\overline{\sigma}(\Lambda) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots = e^{\overline{\sigma}(\Lambda)(t-t_k)} \le e^{\overline{\sigma}(\Lambda)h} = K_1 \tag{C8}$$

where $\overline{\sigma}(\Lambda)$ is the largest singular value of Λ. In general this term can always be bounded since the time difference $t - t_k$ is always smaller than *h*. In other words even when Λ has eigenvalues with positive real part, $\left\| e^{\Lambda(t-t_k)} \right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\| \le K_2 e^{-\alpha_1 k} \tag{C9}$$

with $K_2, \alpha_1 > 0$.
Since k is a function of time we can bound the right term of (C9) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{C10}$$

with $K_3, \alpha > 0$.
So from (C7) using (C8) and (C10) we can conclude:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \le K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\| \tag{C11}$$

Necessity.

We will now proof the necessity part of the theorem by contradiction. Assume the system described by (16) is stable and that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times t_{k+1}^-, in other words, just before the update. Even further we will concentrate on the combined state of the plant $x(t_{k+1}^-)$ and observer $\bar{x}(t_{k+1}^-)$, which are the first two elements of $z(t_{k+1}^-)$.
We will call $\begin{bmatrix} x(t_{k+1}^-) \\ \bar{x}(t_{k+1}^-) \end{bmatrix}$, $\xi(k)$.
Now assume $e^{\Lambda \tau}$ has the following form:

$$e^{\Lambda \tau} = \begin{bmatrix} W_1(\tau) & W_2(\tau) & X_1(\tau) \\ W_3(\tau) & W_4(\tau) & X_2(\tau) \\ Y_1(\tau) & Y_2(\tau) & Z(\tau) \end{bmatrix}$$

For simplicity, lets make the following definitions:

$$W(\tau)=\begin{bmatrix} W_1(\tau) & W_2(\tau) \\ W_3(\tau) & W_4(\tau) \end{bmatrix},\ X(\tau)=\begin{bmatrix} X_1(\tau) \\ X_2(\tau) \end{bmatrix},\ Y(\tau)=\begin{bmatrix} Y_1(\tau) & Y_2(\tau) \end{bmatrix}$$

Then we can express the solution $z(t)$ as:

$$\begin{aligned}
&e^{\Lambda(t-t_k)}\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k z_0 \\
&=\begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix}\begin{bmatrix} (W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} z_0 \\
&=\begin{bmatrix} W(t-t_k)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0
\end{aligned} \tag{C12}$$

Now lets check the values of the solution at times t_{k+1}^-. That is just before the update.

$$z(t_{k+1}^-)=\begin{bmatrix} W(h)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0=\begin{bmatrix} (W(h))^{k+1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 \tag{C13}$$

We also know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first two elements of $z(t_{k+1}^-)$, which we call $\xi(k)$, will in general grow with *k*. In other words we can't ensure $\xi(k)$ will converge to zero for general initial conditions $x_0, \bar{x}_0$.

$$\left\|\begin{bmatrix} x(t_{k+1}^-) \\ \bar{x}(t_{k+1}^-) \end{bmatrix}\right\| = \|\xi(k)\| = \left\|(W(h))^{k+1}\begin{bmatrix} x_0 \\ \bar{x}_0 \end{bmatrix}\right\| \to \infty \qquad as \qquad k \to \infty$$

this clearly means the system cannot be stable, and thus we have a contradiction.

♦

APPENDIX D

Proof of Theorem #5.

On the interval $n \in [n_k, n_{k+1})$, the system response is

$$z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} = \Lambda^{n-n_k} \begin{bmatrix} x(n_k) \\ 0 \end{bmatrix} = \Lambda^{n-n_k} z(n_k) \tag{D2}$$

Now, note that at times n_k, $z(n_k) = \begin{bmatrix} x(n_k) \\ 0 \end{bmatrix}$, that is, the error *e(n)* is reset to zero. We can represent this by

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \breve{z}(n_k)$$

Here $\breve{z}(n_k)$ is the value assumed by $z(n)$ when $n=n_k$ using (D2) for the interval $n \in [n_{k-1}, n_k)$. Using this value of $\breve{z}(n_k)$ we obtain:

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-1})$$

In view of (D2) we have that if at time $n=n_0$, $z(n_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned} z(n) &= \Lambda^{n-n_k} z(n_k) \\ &= \Lambda^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-1}) \\ &= \Lambda^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-2}) \\ &= \Lambda^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-3}) \\ &\ldots \\ &= \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \right)^k z_0 \end{aligned} \tag{D3}$$

Now we know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h$ is of the form $\begin{bmatrix} M & N \\ 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \right)^k$ has the form $\begin{bmatrix} M^K & P \\ 0 & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(n_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ so that

$$\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M^k & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \tag{D4}$$

In view (D4) it is clear that we can represent the system response as:

$$z(n) = \Lambda^{n-n_k}\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k z_0$$
$$n \in [n_k, n_{k+1}),\ with \quad n_{k+1} - n_k = h \tag{D5}$$

♦

APPENDIX E

Proof of Theorem #6.

Sufficiency.

Taking the norm of the solution described as in Theorem #5:

$$\|z(n)\| = \left\|\Lambda^{n-n_k}\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k z_0\right\| \le \left\|\Lambda^{n-n_k}\right\| \cdot \left\|\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k\right\| \cdot \|z_0\| \tag{E7}$$

Now lets analyze the first term on the right hand side of (E7):

$$\left\|\Lambda^{n-n_k}\right\| \le \left(\overline{\sigma}(\Lambda)\right)^{n-n_k} \le \left(\overline{\sigma}(\Lambda)\right)^h = K_1 \tag{E8}$$

where $\overline{\sigma}(\Lambda)$ is the largest singular value of Λ. In general this term can always be bounded since the time difference $n - n_k$ is always smaller than *h*. In other words even when Λ has eigenvalues with positive real part, $\left\|\Lambda^{n-n_k}\right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\|\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k\right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \le K_2 e^{-\alpha_1 k} \tag{E9}$$

with $K_2, \alpha_1 > 0$.
Since k is a function of time we can bound the right term of (E9) in terms of n:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{n-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} n} = K_3 e^{-\alpha n} \tag{E10}$$

with $K_3, \alpha > 0$.
So from (E7) using (E8) and (E10) we can conclude:

$$\|z(n)\| = \left\| \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \le K_1 \cdot K_3 e^{-\alpha n} \cdot \|z_0\| \tag{E11}$$

Necessity.

We will now proof the necessity part of the theorem by contradiction. Assume the system described by (18) is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times n_{k+1}, in other words, just at the update. Even further we will concentrate on a specific term: the state of the plant $x(n_{k+1})$, which is the first element of $z(n_{k+1})$. We will call $x(n_{k+1})$, $\xi(k+1)$.
Now assume Λ^j has the following form:

$$\Lambda^j = \begin{bmatrix} W(j) & X(j) \\ Y(j) & Z(j) \end{bmatrix}$$

Then we can express the solution $z(n_{k+1})$ as:

$$\begin{aligned} & \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \\ & = \begin{bmatrix} W(n-n_k) & X(n-n_k) \\ Y(n-n_k) & Z(n-n_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ & = \begin{bmatrix} W(n-n_k)(W(h))^k & 0 \\ Y(n-n_k)(W(h))^k & 0 \end{bmatrix} z_0 \\ & \forall n \in [n_k, n_{k+1}). \end{aligned} \tag{E12}$$

Now lets check the values of the solution at times n_{k+1}, that is the update time. We know that at this time the error is canceled by the update, and therefore:

$$z(n_{k+1}) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ 0 & 0 \end{bmatrix} z_0 \tag{E13}$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first element of $z(n_{k+1})$, which we call $\xi(k+1)$, will in general grow with *k*. In other words we can't ensure $\xi(k+1)$ will converge to zero for general initial condition x_0.

$$\|x(n_{k+1})\| = \|\xi(k+1)\| = \|(W(h))^{k+1} x_0\| \to \infty \qquad as \quad k \to \infty$$

this clearly means the system cannot be stable, and thus we have a contradiction.

♦

APPENDIX F

Proof of Theorem #7.

On the interval $n \in [n_k, n_{k+1})$, the system response is

$$z(n) = \begin{bmatrix} x(n) \\ \bar{x}(n) \\ e(n) \end{bmatrix} = \Lambda^{n-n_k} \begin{bmatrix} x(n_k) \\ \bar{x}(n_k) \\ 0 \end{bmatrix} = \Lambda^{n-n_k} z(n_k) \tag{F2}$$

Now, note that at times n_k, $z(n_k) = \begin{bmatrix} x(n_k) \\ \bar{x}(n_k) \\ 0 \end{bmatrix}$, that is, the error *e(n)* is reset to zero. We can represent this by

$$z(n_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \breve{z}(n_k)$$

Here $\breve{z}(n_k)$ is the value assumed by *z*(*n*) when *n*=*n*$_k$ using (F2) for the interval $n \in [n_{k-1}, n_k)$. Using this value of $\breve{z}(n_k)$ we obtain:

$$z(n_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-1})$$

In view of (F2) we have that if at time $n=n_0$, $z(n_0)=z_0=\begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$
\begin{aligned}
z(n) &= \Lambda^{n-n_k} z(n_k) \\
&= \Lambda^{n-n_k} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-1}) \\
&= \Lambda^{n-n_k} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-2}) \\
&= \Lambda^{n-n_k} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h z(n_{k-3}) \\
&\ldots \\
&= \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \right)^k z_0
\end{aligned}
\tag{F3}
$$

Now we know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h$ is of the form $\begin{bmatrix} M_1 & M_2 & N_1 \\ M_3 & M_4 & N_2 \\ 0 & 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \right)^k$ has the form $\begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(n_0)=z_0=\begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix}$ so that

$$
\begin{aligned}
\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k \begin{bmatrix} x_0 \\ \bar{x}_0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} \\
= \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \bar{x}_0 \\ 0 \end{bmatrix} &= \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{F4}
$$

In view (F4) it is clear that we can represent the system response as:

$$z(n) = \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0$$
$$n \in [n_k, n_{k+1}),\ with\ \ n_{k+1} - n_k = h \tag{F5}$$

♦

APPENDIX G

Proof of Theorem #8.

Sufficiency.

Taking the norm of the solution described as in Theorem #7:

$$\|z(n)\| = \left\| \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \le \left\| \Lambda^{n-n_k} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\| \tag{G7}$$

Now lets analyze the first term on the right hand side of (G7):

$$\left\| \Lambda^{n-n_k} \right\| \le \left(\overline{\sigma}(\Lambda) \right)^{n-n_k} \le \left(\overline{\sigma}(\Lambda) \right)^h = K_1 \tag{G8}$$

where $\overline{\sigma}(\ln(\Lambda))$ is the largest singular value of $\ln(\Lambda)$. In general this term can always be bounded since the time difference $n - n_k$ is always smaller than *h*. In other words even when Λ has eigenvalues with positive real part, $\left\| \Lambda^{n-n_k} \right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 e^{-\alpha_1 k}$$

(G9)

with $K_2, \alpha_1 > 0$.

Since k is a function of time we can bound the right term of (G9) in terms of n:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{n-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} n} = K_3 e^{-\alpha n}$$

(G10)

with $K_3, \alpha > 0$.

So from (G7) using (G8) and (G10) we can conclude:

$$\|z(n)\| = \left\| \Lambda^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_3 e^{-\alpha n} \cdot \|z_0\|$$

(G11)

Necessity.

We will now proof the necessity part of the theorem by contradiction. Assume the system described by (19) is stable and that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times n_{k+1}, in other words, just at the update. Even further we will concentrate on the combined state of the plant $x(n_{k+1})$ and observer $\bar{x}(n_{k+1})$, which are the first two elements of $z(n_{k+1})$.

We will call $\begin{bmatrix} x(n_{k+1}) \\ \bar{x}(n_{k+1}) \end{bmatrix}$, $\xi(k+1)$.

Now assume Λ^j has the following form:

$$\Lambda^j = \begin{bmatrix} W_1(j) & W_2(j) & X_1(j) \\ W_3(j) & W_4(j) & X_2(j) \\ Y_1(j) & Y_2(j) & Z(j) \end{bmatrix}$$

For simplicity, lets make the following definitions:

$$W(j)=\begin{bmatrix} W_1(j) & W_2(j) \\ W_3(j) & W_4(j) \end{bmatrix},\ X(j)=\begin{bmatrix} X_1(j) \\ X_2(j) \end{bmatrix},\ Y(j)=\begin{bmatrix} Y_1(j) & Y_2(j) \end{bmatrix}$$

Then we can express the solution $z(n_{k+1})$ as:

$$\begin{aligned}
&\Lambda^{n-n_k}\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k z_0 \\
&=\begin{bmatrix} W(n-n_k) & X(n-n_k) \\ Y(n-n_k) & Z(n-n_k) \end{bmatrix}\begin{bmatrix} (W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} z_0 \\
&=\begin{bmatrix} W(n-n_k)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(n-n_k)(W(h))^k & 0 \end{bmatrix} z_0 \\
&\forall n\in[n_k,n_{k+1}).
\end{aligned} \tag{G12}$$

Now lets check the values of the solution at times n_{k+1}, that is the update time. We know that at this time the error is canceled by the update, and therefore:

$$z(n_{k+1})=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} W(h)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(h)(W(h))^k & 0 \end{bmatrix}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} z_0=\begin{bmatrix} (W(h))^{k+1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} z_0 \tag{G13}$$

We also know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\Lambda^h\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first two element of $z(n_{k+1})$, which we call $\xi(k+1)$, will in general grow with *k*. In other words we can't ensure $\xi(k+1)$ will converge to zero for general initial condition x_0.

$$\left\|\begin{bmatrix} x(n_{k+1}) \\ \bar{x}(n_{k+1}) \end{bmatrix}\right\|=\|\xi(k+1)\|=\left\|(W(h))^{k+1}x_0\right\|\to\infty \quad as \quad k\to\infty$$

this clearly means the system cannot be stable, and thus we have a contradiction.

♦

APPENDIX H

Proof of Theorem #9.

Assume the system starts at time t_0 with initial conditions $z(t_0)=\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$. On the interval $t\in[t_0,t_1-\tau)$, the system response is:

$$z(t)=\begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix}=e^{\Lambda(t-t_0)}\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}.$$

At $t=(t_1-\tau)^-$:

$$z((t_1-\tau)^-)=\begin{bmatrix} x((t_1-\tau)^-) \\ \breve{e}((t_1-\tau)^-) \\ \hat{e}((t_1-\tau)^-) \end{bmatrix}=e^{\Lambda(h-\tau)}\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}.$$

According to the update law, at $t=t_1-\tau$, $\breve{e}\leftarrow 0$ and $\hat{e}\leftarrow x-\hat{x}=\breve{e}+\hat{e}$, so:

$$z(t_1-\tau)=\begin{bmatrix} x(t_1-\tau) \\ \breve{e}(t_1-\tau) \\ \hat{e}(t_1-\tau) \end{bmatrix}=\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix}e^{\Lambda(h-\tau)}\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}.$$

Continuing with the interval $t\in[t_1-\tau,t_1)$

$$z(t)=\begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix}=e^{\Lambda(t-t_1+\tau)}\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix}e^{\Lambda(h-\tau)}\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$$

At $t=t_1^-$:

$$z(t_1^-)=\begin{bmatrix} x(t_1^-) \\ \breve{e}(t_1^-) \\ \hat{e}(t_1^-) \end{bmatrix}=e^{\Lambda\tau}\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix}e^{\Lambda(h-\tau)}\begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}.$$

Now, according to the update law, at $t = t_1$, $\hat{e} \leftarrow 0$, so:

$$z(t_1) = \begin{bmatrix} x(t_1) \\ \breve{e}(t_1) \\ \hat{e}(t_1) \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}.$$

It is easy to see that the response is in general:

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$$

for $t \in [t_k, t_{k+1} - \tau)$

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix} = e^{\Lambda(t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$$

for $t \in [t_{k+1} - \tau, t_{k+1})$.

(H1)

It becomes evident here that if the matrix $M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}$ has its eigenvalues inside the unit circle the system will be globally asymptotically stable.

Sufficiency.

Taking the norm of the solutions described in (H1):

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k z_0 \right\|$$

$$\leq \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k \right\| \cdot \|z_0\|$$

for $t \in [t_k, t_{k+1} - \tau)$

$$\|z(t)\| = \left\| e^{\Lambda(t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k z_0 \right\|$$

$$\leq \left\| e^{\Lambda(t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^k \right\| \cdot \|z_0\|$$

for $t \in [t_{k+1} - \tau, t_{k+1})$.

(H7)

Now lets analyze the first terms on the right hand side of the inequalities in (H7):

$$\left\| e^{\Lambda(t-t_k)} \right\| \leq 1 + (t-t_k)\overline{\sigma}(\Lambda) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots = e^{\overline{\sigma}(\Lambda)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda)h} = K_a$$

(H8.1)

$$\left\| e^{\Lambda(t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right\| \leq \left\| e^{\Lambda(t-t_{k+1}+\tau)} \right\| \cdot \left\| e^{\Lambda(h-\tau)} \right\| = \left\| e^{\Lambda(t-t_{k+1}+\tau)} \right\| \cdot C$$

$$\leq \left(1 + (t - t_k + \tau)\overline{\sigma}(\Lambda) + \frac{(t-t_k+\tau)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots \right) \cdot C$$

$$= e^{\overline{\sigma}(\Lambda)(t-t_{k+1}+\tau)} \cdot C \leq e^{\overline{\sigma}(\Lambda)\tau} \cdot C = K_b$$

(H8.2)

where $\overline{\sigma}(\Lambda)$ is the largest singular value of Λ. We define a new constant K_1=max(K_a,K_b).

We now study the term $\left\|\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}\right)^k\right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}$ lie inside the unit circle:

$$\left\|\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}\right)^k\right\| \leq K_2 e^{-\alpha_1 k} \tag{H9}$$

with $K_2, \alpha_1 > 0$.

Since k is a function of time we can bound the right term of (H9) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{H10}$$

with $K_3, \alpha > 0$.

So from (H7) using (H8.1), (H8.2), and (H10) we can conclude:

$$\|z(t)\| = \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\| \tag{H11}$$

Necessity.

We will now proof the necessity part of the theorem by contradiction. Assume the system described by (20) and (21) is stable and that $M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times t_{k+1}^-. We can express the solution $z(t_{k+1}^-)$ as:

$$
\begin{aligned}
z(t_{k+1}^{-}) &= \xi(k) \\
&= e^{\Lambda(\tau)}\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} \right)^{k} z_0 \\
&= e^{\Lambda(\tau)}\begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda(h-\tau)} M^{k} z_0
\end{aligned}
\tag{H12}
$$

We also know that M has at least one eigenvalue outside the unit circle. This means that $z(t_{k+1}^{-})$ will in general grow with k. In other words we can't ensure $\xi(k)$ will converge to zero for general initial condition z_0.

$$\left\| z(t_{k+1}^{-}) \right\| = \left\| \xi(k) \right\| \to \infty \qquad \text{as} \qquad k \to \infty$$

this clearly means the system cannot be stable, and thus we have a contradiction.

♦

8 REFERENCES

[1] R. Brockett, "Minimum Attention Control," *Proceedings of the 36th Conference on Decision and Control*, 1997, pp. 2628-2632.

[2] N. Elia and S. Mittler., "Stabilization of Linear Systems With Limited Information," *IEEE Transactions on Automatic Control*, 2001, pp. 1384-1400.

[3] P. Bauer, M. Schitiu, C. Lorand, and K. Premaratne, "Total Delay Compensation in LAN Control Systems and Implications for Scheduling," *Proceeding of the American Control Conference*, June 2001, pp. 4300-4305.

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

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

[6] G. Stewart, "Introduction to Matrix Computations", 1st edition, Academic Press, New York, 1973

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

[8] G. Walsh and H. Ye, "Scheduling of Networked Control Systems," *IEEE Control Systems Magazine*, February 2001, Vol. 21, No. 1.

[9] G. Dullerud and S. Lall, "Asynchronous Hybrid Systems With Jumps- Analysis And Synthesis Methods," *Systems & Control Letters*, June 1999, Vol. 37, Issue 2, pp. 61-69.

[10] H. Rehbinder and M. Sanfridson, "Scheduling Of A Limited Communication Channel For Optimal Control," *Proceedings of the Conference on Decision and Control*, 2000, pp. 1011-1016.

[11] H. Ishii and B. Francis, "Stabilization With Control Networks," *Control 2000, Cambridge UK*, Sept 4-7, 2000

[12] B. Bamieh, J. Boyd Pearson, B. Francis, and A. Tannenbaum, "A Lifting Technique For Linear Periodic Systems With Applications To Sampled-Data Control," *Systems & Control Letters*, 1991, Vol 12, pp.79-88.

[13] L. Mirkin, H. Rotstein, and Z. Palmor, "H_2 and H_{inf} Design Of Sampled-Data Systems Using Lifting. Part I: General Framework And Solutions," *SIAM Journal Of Control And Optimization*, Vol 38, No. 1, pp. 175-196.

[14] O. Beldiman, G. Walsh, L. Bushnell, "Predictors For Networked Control Systems," *Proceedings Of The American Control Conference*, June 2000, pp. 2347-2351.

[15] H. Ishii and B. Francis, "Stabilizing a Linear System by Switching Control and Output Feedback with Dwell Time," *Proceedings Of The 40th IEEE Conference On Decision And Control*, December 2001, pp. 191-196.

[16] G. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic Behavior of Networked Control Systems," *Proceedings of the International Conference on Control Applications*, August 1999.

[17] O. Beldiman, L. Bushnell, G. Walsh, H.O. Wang, and Y. Hong, "Perturbations in Networked Control Systems," *Proceedings of ASME-IMECE*, November 2001.

[18] G. Walsh, O. Beldiman, and L. Bushnell, "Error Encoding Algorithms for Networked Control Systems," *Proceedings of the IEEE Conference on Decision and Control*, December 1999, pp. 4933-4939.

[19] A. Stubbs, and G.E. Dullerud, "Networked Control of Distributed Systems," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 203-204.

[20] I.R. Petersen, and A.V. Savkin, "Multi-rate Stabilization of Multivariable Discrete-Time Linear Systems via a Limited Capacity Communication Channel," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 304-309.

[21] A.S. Matveev, A.V. Savkin, "Optimal State Estimation in Networked Systems with Asynchronous Communication Channels and Switched Sensors**,"** *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 825-830.

[22] L. Xiao, M. Johansson, H. Hindi, S. Boyd, A. Goldsmith, "Joint Optimization of Communication Rates and Linear Systems," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 2321-2326.

[23] A.S. Matveev, A.V. Savkin, "Optimal Control of Networked Systems via Asynchronous Communication Channels with Irregular Delays," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 2323-2332.

[24] D. Hristu-Varsakelis, "Feedback Control Systems as Users of a Shared Network: Communication Sequences that Guarantee Stability," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 3631-3636.