Chapter 2
Model-Based Control Systems: Stability

In this book we study networked control systems that make explicit use of existing knowledge of the plant dynamics, encapsulated in the mathematical model of the plant, to enhance the performance of the system. This class of networked systems is called Model-Based Networked Control Systems (MB-NCS). The performance of a networked control system depends on the performance of the communication network in addition to traditional control systems performance measures. The bandwidth of the communication network used by the control system is of major concern, since other control and data acquisition systems will typically be sharing the same digital communication network.

It turns out that stability margins, controller robustness, and other stability and performance measures may be significantly improved when knowledge of the plant dynamics is explicitly used to predict plant behavior. Note that the plant model is always used to design controllers in standard control design. The difference here is that the plant model is used explicitly in the controller implementation to great advantage. This is possible today because existing inexpensive computation power allows the simulation of the model of the plant in real time.

In this chapter we lay the foundations for the type of networked architecture that we call MB-NCS. We also provide a thorough analysis of the behavior of the system. The focus of the chapter is in obtaining conditions that result in a stable networked system. Stabilization depends on the chosen control gain, the accuracy of the model, and the update interval. We derive necessary and sufficient stabilizing conditions for both continuous and discrete-time linear time-invariant systems. In the following chapters we significantly extend these results to consider different scenarios, for instance, we include the more general case when only a linear combination of the states (system output) is available for measurement and we also consider the effect of network induced delays when state feedback is possible.

The contents of this chapter are as follows: In Sect. 2.1 the Model-Based Control architecture is introduced. In Sect. 2.2 the continuous-time case with state feedback is considered and in Sect. 2.3 the discrete-time case is addressed. Alternative stability conditions are offered in Sect. 2.4. Notes and references are in Sect. 2.5.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_2,

2.1 Fundamentals: Model-Based Control Architecture

We consider here the control of a linear time-invariant dynamical system where the state sensors are connected to controllers/actuators via a network; more complex models are considered in later chapters. The main goal is to reduce 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 the stabilization of the plant even under slow network conditions. Although in principle, we can use the same framework to study the problem of packet dropouts in NCS, the aim here is to purposely avoid frequent broadcasting of unnecessary sensor measurements so to reduce traffic in the network as much as possible, which in turn reduces the presence of the problems associated with high network load such as packet collisions and network induced delays.

We will concentrate on characterizing the time interval between successive transmissions of data from the sensor to the controller/actuator (update intervals of the state of the model); Transmission of data from sensor to controller and from controller to actuator is considered later in the book. *Our goal in this chapter will be to identify the maximum update intervals that can be used to transmit measurement updates between the sensor and the actuator while keeping the system stabl*e. 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 update time we will use the knowledge we have of the plant dynamics. The plant model is used at the controller/actuator side to approximate the plant behavior. The sensor is able to reduce the rate at which it transmits data, since the model can provide information to generate appropriate control inputs while the systems is running open loop. *Note that in standard digital control a zero-order-hold keeps the input constant in between samples. Here the input between samples is calculated based on the plant model and it is reasonable to assume that it will be better suited than the constant input.*

The main idea is to update the state of the model using the actual state of the plant 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 control architecture is shown in Fig. 2.1.

In our control architecture 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 closed-loop information needed to overcome model uncertainties and plant disturbances. In the remaining of this chapter we assume that all states can be measured and these measurements can be transmitted through the network to update the model of the system in the controller node. The more general case when only a linear combination of the states can be sensed is studied in the following chapter together with the network induced delays case. In this chapter we provide necessary and sufficient conditions for stability that result in a maximum update time, which depends mainly on the model inaccuracies but also on the designed control gain.

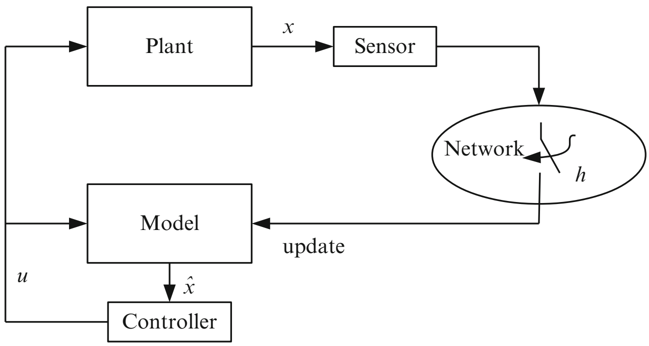

Fig. 2.1 The model-based networked control system (MB-NCS) architecture

2.2 Continuous-Time LTI Systems: State Feedback

In this section, we introduce the foundations of our Model-Based approach. We consider multi-input, multi-output linear time-invariant continuous-time systems and their state variable representations, and we assume a constant linear state feedback control law. Necessary and sufficient conditions are derived for the stability of the compensated system in Theorem 2.3, the main result of the section. Illustrative examples are also included. Discrete-time systems are considered in the next section and output feedback in the next chapter.

If all the states are available for measurement, then the sensors can send this information through the network to update the model's vector state. We will assume that the compensated model is stable, which is typical in control systems, 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. The goal is to find the largest constant update period at which the network must update the model state in the controller for stability, that is, we are seeking an upper bound for h the update time. Usual assumptions in the literature include requiring a stable plant or in the case of a discrete controller, a smaller update time than the sampling time. Here we do not make any of these assumptions. The original plant may be open-loop unstable.

Consider the control system of Fig. 2.1 where the plant, the plant model and the controller are described by:

$$\begin{aligned} \text{Plant}: \quad & \dot{x} = Ax + Bu, \\ \text{Plant model}: \quad & \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u, \\ \text{Controller}: \quad & u = K\hat{x}. \end{aligned} \tag{2.1}$$

Since the sensor has the full state vector available, its function will be to send the state information through the network every h seconds. The state error is defined as:

$$\text{State Error}: \quad e = x - \hat{x}, \tag{2.2}$$

and represents the difference between the plant state and the model state. The modeling error matrices: $\widetilde{A} = A - \hat{A}$ and $\widetilde{B} = B - \hat{B}$ represent the difference between the plant and the model. The periodic update time instants are denoted by t_k, where

$$t_k - t_{k-1} = h \quad \text{for} \quad k = 1, 2, \ldots \tag{2.3}$$

(here, h is a constant). The choice of h, being a constant, is simple to implement and also results in a simple analysis procedure as shown below. Update intervals that are not constant will be considered as well; in Chap. 5, we address time-varying update intervals and in Chap. 6 we study event (error)-based update intervals.

Since the model state is updated every t_k s,

$$e(t_k) = 0 \quad \text{for} \quad k = 1, 2, \ldots,. \tag{2.4}$$

This resetting of the state error at every update time instant is a key characteristic of our control system. Now for $t \in [t_k, t_{k+1})$, we have that: $u = K\hat{x}$ so the overall system is described by

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{2.5}$$

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

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

$$\begin{gathered} \begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \\ \begin{bmatrix} x(t_k) \\ e(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ 0 \end{bmatrix}, \\ \forall t \in [t_k, t_{k+1}), \quad \text{with} \quad t_{k+1} - t_k = h. \end{gathered} \tag{2.6}$$

Define the augmented state vector $z(t)$ and the open-loop augmented state matrix Λ:

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \tag{2.7}$$

$$\Lambda = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \tag{2.8}$$

so that (2.6) can be rewritten as

$$\dot{z} = \Lambda z \quad \text{for} \quad t \in [t_k, t_{k+1}). \tag{2.9}$$

We will now express $z(t)$ in terms of the initial condition $x(t_0)$. Then we will show under what conditions the system will be stable.

Proposition 2.1 *The system described by* (2.6) *with initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ 0 \end{bmatrix} = z_0$ *has the following response:*

$$z(t) = e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0$$
$$t \in \left[t_k, t_{k+1}\right), \quad \text{with} \quad t_{k+1} - t_k = h. \tag{2.10}$$

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

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix} = e^{\Lambda(t-t_k)} z(t_k). \tag{2.11}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix}$, that is, the error is reset to 0, that is, $e(t_k) = 0$ for $k = 1,2,\ldots,$. We can represent this by

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z\left(t_k^-\right). \tag{2.12}$$

Using (2.11) to calculate $z(t_k^-)$ we obtain

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}). \tag{2.13}$$

In view of (2.11) 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:

$$\begin{aligned} z(t) &= e^{\Lambda(t-t_k)} z\left(t_k\right) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-2}) \\ &= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k z_0 \end{aligned} \tag{2.14}$$

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:

$$\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix}. \quad (2.15)$$

In view of (2.15) it is clear that we can represent the system response as in (2.10):

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

♦

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

Definition 2.2 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

$$\|\phi(t, t_0, z_0)\| \leq k(\beta)\|z_0\|e^{-\alpha(t-t_0)}, \qquad \forall t \geq t_0 \quad (2.16)$$

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

With this definition of stability we state the following theorem characterizing the necessary and sufficient conditions for the system described by (2.6) 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 2.3 *The system described by* (2.6) *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 in (2.10), (in Proposition 2.1):

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\|$$
$$\leq \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\|. \tag{2.17}$$

Now let us analyze the first term on the right-hand side of (2.17):

$$\left\| e^{\Lambda(t-t_k)} \right\| \leq 1 + (t-t_k)\overline{\sigma}(\Lambda) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots = e^{\overline{\sigma}(\Lambda)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda)h}$$
$$= K_1 \tag{2.18}$$

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:

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 e^{-\alpha_1 k} \tag{2.19}$$

with $K_2, \alpha_1 > 0$. Since k is a function of time, we can bound the right term of (2.19) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{2.20}$$

with $K_3,\ \alpha > 0$. So from (2.17) using (2.18) and (2.20) we can conclude:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\|. \tag{2.21}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the state feedback MB-NCS 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, the sequence of periodic samples of the response should converge to 0 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:

$$e^{\Lambda\tau} = \begin{bmatrix} W(\tau) & X(\tau) \\ Y(\tau) & Z(\tau) \end{bmatrix}. \tag{2.22}$$

In view of (2.11) we can express the response $z(t)$ as:

$$\begin{aligned} & e^{\Lambda(t-t_k)}\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k z_0 \\ &= \begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(t-t_k)(W(h))^k & 0 \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0. \end{aligned} \tag{2.23}$$

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

$$z\left(t_{k+1}^-\right) = \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0. \tag{2.24}$$

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 implies that the first element of $z(t_{k+1}^-)$, which we call $\xi(k)$, will in general grow with k. In other words we cannot ensure $\xi(k)$ will converge to 0 for general initial condition x_0. That is

$$\left\|x\left(t_{k+1}^-\right)\right\| = \|\xi(k)\| = \left\|(W(h))^{k+1}x_0\right\| \to \infty \quad \text{as} \quad k \to \infty \tag{2.25}$$

which implies that the system cannot be stable, and thus we have a contradiction. ♦

Example 2.1 Consider the following unstable system (plant) dynamics:

$$A = \begin{bmatrix} 0.14 & 1.25 \\ 0 & 0.08 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1.07 \end{bmatrix}. \tag{2.26}$$

The exact parameters of the plant are unknown. In general, a model containing nominal parameters of the real parameters is available. Assume that (2.26) is modeled using the following nominal parameters

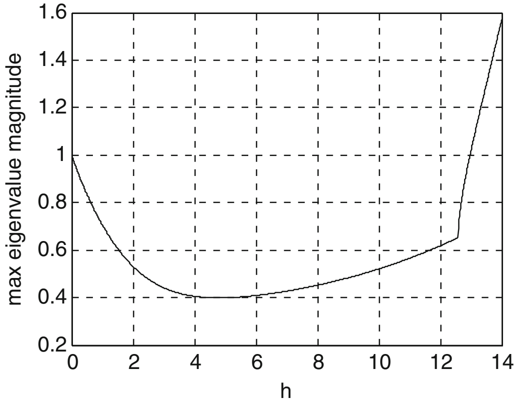

Fig. 2.2 Maximum eigenvalue magnitude of the test matrix M vs. the update time h

$$\hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.27}$$

We will use the state feedback controller with control gain given by $K = [-0.2 \ \ -0.9]$. We now search for the largest 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. Figure 2.2 shows the eigenvalue with maximum magnitude of this matrix. From this search we can use values of $h < 12.96$ s in order to design a stable networked system using the model-based approach. Figure 2.3 shows the response for different values of h and the same initial conditions $x_0 = [1 \ \ -1]^T$, while the model initial conditions are set equal to 0. Note that the time scale is different for the responses shown in the second column in order to show that the response is stable for $h = 12.9$ s and unstable for $h = 13$ s, as expected.

In order to draw a comparison with other approaches commonly used in networked control, we now use a model of the system equivalent to a zero-order-hold (ZOH) model. This is also the same type of implementation used in traditional sampled-data control systems. Here, the ZOH holds the value of the most recent measurement constant until a new measurement arrives to update the controller. We can use the results of this section to find the largest possible value of the update period under this scenario since the ZOH is just a special case of our approach that can be easily modeled using the following:

$$\hat{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.28}$$

From Fig. 2.4 we can see that the admissible values of the update period that preserve stability are $h < 2.13$. Figure 2.5 shows the response for two choices $h = 1.5$ (stable) and $h = 2.15$ (unstable). The initial conditions are the same in both cases. We can see in this example that there is a significant difference in the range of values for the update periods when using the ZOH model (2.28) and using

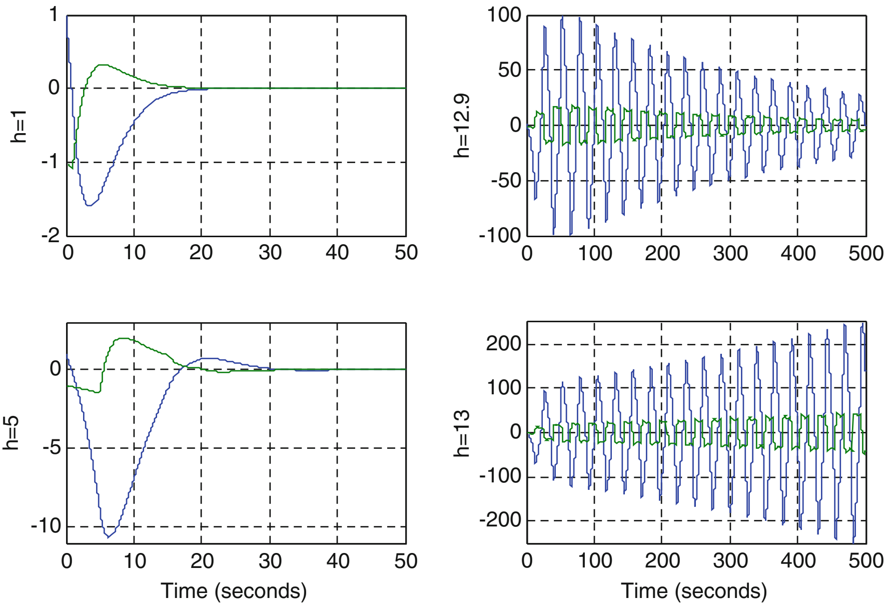

Fig. 2.3 System response for different values of h. For $h = 1$, $h = 5$, and $h = 12.9$ s the system is stable. For $h = 13$ s the system is unstable

the nominal model (2.27). Even though the nominal model does not represent the exact dynamics of the system, it provides a more accurate estimation of the real state between updates that allows for an implementation of the networked system with longer update periods. This is a clear benefit of the model-based approach in reducing the necessary bandwidth for stability.

Example 2.2 Applicability of State Feedback MB-NCS Results. Regarding applicability to real world cases, we see that, as stated in Theorem 2.3, the stability properties of a networked control system can be determined by studying the eigenvalues of the matrix:

$$M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \text{ where } \Lambda = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}. \tag{2.29}$$

The matrix Λ can also be expressed in terms of the model and the errors only:

$$\Lambda = \begin{bmatrix} \hat{A} + \widetilde{A} + \left(\hat{B} + \widetilde{B}\right)K & -\left(\hat{B} + \widetilde{B}\right)K \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}. \tag{2.30}$$

In real applications the state-space model is usually obtained by studying the structure and behavior of the plant. The uncertainties can frequently be expressed as

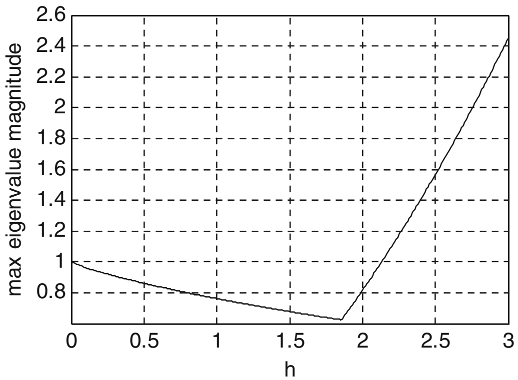

Fig. 2.4 Maximum eigenvalue magnitude of the test matrix M vs. the update time h for the ZOH model

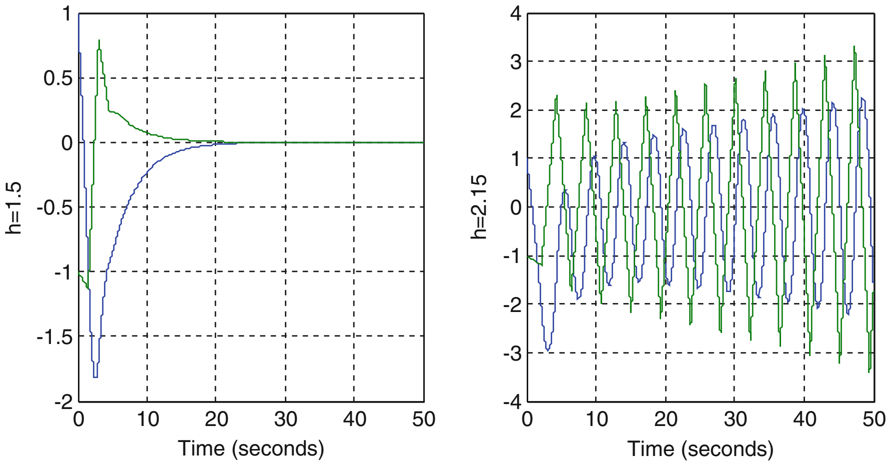

Fig. 2.5 System response for different values of h using a ZOH model. For $h = 1.5$ s the system is stable. For $h = 2.15$ s the system is unstable

tolerances over the different measured parameter values of the plant. This can be mapped into structured or parametric uncertainties on the state-space matrices. The following is an example that provides insight on how the theorem can be applied if two entries on the A matrix of the model can vary within a certain interval.

$$\begin{aligned} &\text{model}: \quad \hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\ &\text{plant}: \quad A = \begin{bmatrix} 0 & 1 + \widetilde{a}_{12} \\ 0 + \widetilde{a}_{21} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\ &\qquad\qquad \text{with } \widetilde{a}_{12} = [-0.5, 0.5], \widetilde{a}_{21} = [-0.5, 0.5] \\ &\text{controller}: K = [-1, \ -2]. \end{aligned} \tag{2.31}$$

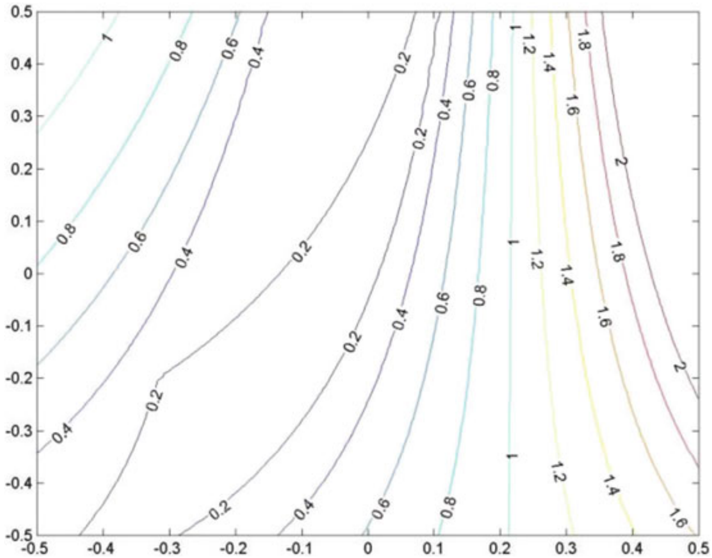

Fig. 2.6 Contour plot maximum eigenvalue. Magnitude vs. model error

The system will now be tested for an update time of $h = 2.5$ time units. Figure 2.6 represents a contour plot where the contour of height equal to 1 separates the stable and unstable regions. In Fig. 2.7 we have plotted the surface representing the maximum eigenvalue magnitude for the test matrix M as a function of the (1,2) and (2,1) entries and for a given selection of values of those entries.

It is easy to isolate the stable and unstable regions in the uncertainty parameter plane. The stable region is between the lines labeled as 1 in Fig. 2.6. The same procedure can be used in real applications to verify the stability of a networked control system with a certain model and update time. The procedure can be repeated for different values of uncertainties on as many elements of A as needed and for different update intervals as well.

Example 2.3 In this example we consider the instrument servo (dc motor driving an inertial load) dynamics from Example 6A in [77]

$$\begin{bmatrix} \dot{e} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix} \begin{bmatrix} e \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} u \tag{2.32}$$

where $e = \theta - \theta_r$ represents the error between the current angular position θ and the desired position θ_r, where the desired position is assumed to be constant, ω is the angular velocity, and u is the applied voltage. The parameters α and β represent constants that depend on the physical parameters of the motor and load.

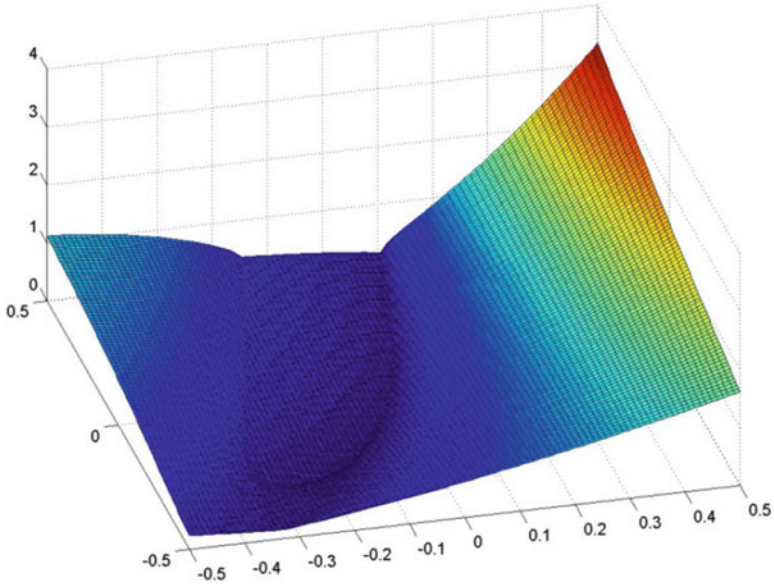

Fig. 2.7 Maximum eigenvalue magnitude vs. model error

The nominal parameters are $\hat{\alpha} = 1$, $\hat{\beta} = 3$. The real parameters are given by $\alpha = 0.75$, $\beta = 2.58$. The model parameters are used for controller design: $K = [-0.1667 \ -0.1667]$. According to the eigenvalue search described in this section, the networked system is stable for any value $h > 0$; however the transient response will significantly differ for different choices of the update period. Figure 2.8 shows the response for two different update periods where we can see that the networked system converges more rapidly for $h = 1$ s than for $h = 10$ s.

Remark Models in which uncertainty is represented in terms of norms as a ball around the model can also be derived, but the conditions for stability would only be sufficient. The results in this section then offer a less conservative approach that can be readily applied to real applications as well. The conditions in Theorem 2.3 assume that noiseless measurements are transmitted at the update instants t_k. In Chap. 12 we consider noisy measurements within the MB-NCS framework.

Remark Optimization problem. The design of optimal controllers for MB-NCS represents an interesting research problem. In addition to optimizing the response of the control systems and penalizing the excessive use of control effort, we should also weight the use of network resources. The overall problem requires not only the design of the optimal control law but also the design of the optimal scheduling law.

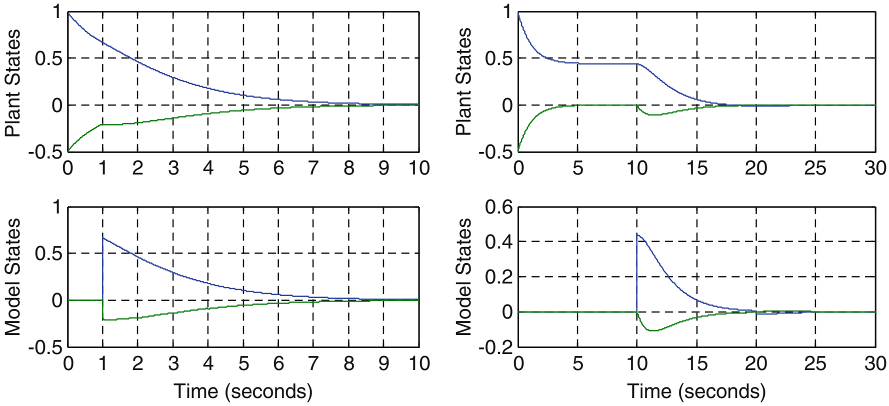

Fig. 2.8 Plant and model states. *Left*: $h = 1$ s. *Right*: $h = 10$ s

The scheduling law can, in general, be a time-varying transmission policy. Chapter 9 addresses this type of problem by using the following cost function:

$$\min_{u,\beta} J = \frac{1}{2}x^T(T)Gx(T) + \int_0^T \frac{1}{2}\left(x^T(t)Qx(t) + u^T(t)Ru(t)\right)dt \\ + \text{cost of transmission.} \tag{2.33}$$

Note that this problem considers the uncertain nature of the model parameters with respect to the real parameters of the system which significantly increases the complexity.

Remark 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 $\widetilde{A}$ and $\widetilde{B}$. 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 on Λ we obtain:

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

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.

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

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

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

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

$$
N = e^{\left(\hat{A}+\hat{B}K\right)h} + e^{Ah}\int_0^h e^{-A\tau}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau
$$

Then the eigenvalues of $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ are inside the unit circle if and only if the eigenvalues of N are inside the unit circle. One can gain a better insight of the system by observing the structure of N. To start with, we observe that the eigenvalues of the compensated model appear in the first term of N. In that sense we can see the term $\Delta = e^{Ah}\int_0^h e^{-A\tau}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$ as a perturbation over the desired eigenvalues. The important variables in the perturbation term Δ are the uncertainties and the update intervals. In general, for unstable open-loop systems, the perturbation Δ increases as either the update intervals or the model uncertainties increase. On the other hand, even if the eigenvalues of the original plant were unstable the perturbation Δ can be made small enough by having $\widetilde{A}+\widetilde{B}K$ small and thus minimizing their impact over the eigenvalues of the compensated plant.

Remark An important discussion concerning the control of continuous-time systems using the MB-NCS framework is the following. When the update interval h is getting small and approaches 0, then according to the analysis just described in this section, the model/controller tries to update its internal state very fast or in other words it tries to switch infinitely many times in finite intervals of time without letting the model to execute and estimate the plant state. If we compute the eigenvalues of the matrix given in Theorem 2.3 (or the eigenvalues of N in the previous remark) for $h = 0$ we obtain eigenvalues equal to one which represent this behavior of attempting to update the same value over and over again. However, in a physical continuous-time system we cannot update the controller infinitely fast in real time (compare to a typical sampled-data system in the same situation) and undesired system response may occur. The case when $h = 0$ may also be considered in a different way. The case when $h = 0$ corresponds to the controller receiving continuous feedback measurements from the sensor; and instead of updating the model many times we can use directly the plant measurements to control the system. We may also consider the networked system as operating in two modes: In closed-loop mode, that is, when the controller receives continuous feedback from the sensor ($h = 0$), the model is unnecessary and the feedback information is immediately used by the controller to compute the control input. In open-loop mode, the model is now used to estimate the state of the plant using the last received feedback value of the real state as the initial condition. We will study this case in more detail in Chap. 4.

2.3 Discrete-Time LTI Systems: State Feedback

In this section, we present results for the discrete-time case. They are analogous to the continuous-time results of the previous section. We consider multi-input, multi-output linear time-invariant discrete-time systems and their state variable representations, and we assume constant linear feedback control law. Necessary and sufficient conditions are derived for the stability of the compensated systems in Theorem 2.6, the main result of the section. Illustrative examples are included. Output feedback is considered in the next chapter.

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

$$x(n+1) = Ax(n) + Bu(n) \tag{2.34}$$

with model

$$\hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n) \tag{2.35}$$

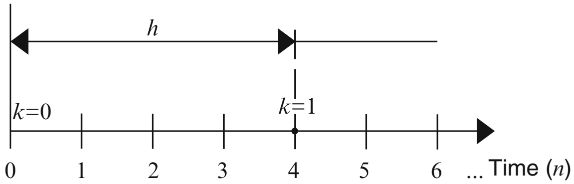

Fig. 2.9 Representation of the state updates for the discrete-time case

where $n = 0,1,2,\ldots$ The control input and the state error are defined, respectively, by:

$$\begin{aligned} u(n) &= K\hat{x}(n) \\ e(n) &= x(n) - \hat{x}(n). \end{aligned} \tag{2.36}$$

There are some assumptions we need to make before we carry our results over to the discrete-time domain. First, in order to have consistency in updates from the sensor side to the actuator side, we must ensure that both the sensor and the actuator/controller are synchronized in the sense that both will carry out their respective tasks at the same sampling time. Moreover state updates will occur only at some of these sampling times. This implies that the update interval h will be an integer number, representing the number of time units between updates of the actuator's model. For example, if $h = 4$, see Fig. 2.9, then the sensor will send measurement updates every 4 time units as indexed by n.

A slightly different way to explain these ideas is by considering two sampling periods. One of them occurs at time instants indexed by n and corresponds to the sampling period of the original plant (2.34). The other one occurs at time instants indexed by hn (for positive integer h) and corresponds to the selected update period at which the sensor sends information to the controller in order to update the state of the model. When the update intervals are constant the networked system can be seen as a linear periodic system. When $h = 1$ the controller is updated at each time index n of the system's clock, i.e., the discrete-time system will operate in closed-loop mode. In Sect. 2.4 we pursue this idea further, but for the moment we will focus on the first explanation provided on the previous paragraph.

The approach that we are going to follow to determine the stability of the networked system is analogous to the one used for continuous plants. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described by:

$$\begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}, \tag{2.37}$$
$$\text{for } n \in [n_k, n_{k+1}), \quad n_{k+1} - n_k = h, \quad \text{and} \quad e(n_{k+1}) = 0.$$

We also use the definitions (2.7) and (2.8) to express (2.37) in compact form as follows:

$$z(n+1) = \Lambda_D z(n) \text{ for } n \in [n_k, n_{k+1}), \quad \text{and} \quad z(n_{k+1}) = \begin{bmatrix} x(n_{k+1}) \\ 0 \end{bmatrix}. \tag{2.38}$$

where

$$z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}$$

$$\Lambda_D = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}$$

Proposition 2.4 *The system described by* (2.38) *with initial condition* $z(n_0) = \begin{bmatrix} x(n_0) \\ 0 \end{bmatrix} = z_0$, *has the response:*

$$z(n) = \Lambda_D{}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, \quad \text{for } n \in [n_k, n_{k+1}), \quad n_k - n_{k+1} = h. \tag{2.39}$$

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

$$z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} = \Lambda_D{}^{n-n_k} \begin{bmatrix} x(n_k) \\ 0 \end{bmatrix} = \Lambda_D{}^{n-n_k} z(n_k). \tag{2.40}$$

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 0. We can represent this by

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \breve{z}(n_k). \tag{2.41}$$

Here $\breve{z}(n_k)$ is the value assumed by $z(n)$ when $n = n_k$ using (2.40) for the interval $n \in [n_{k-1}, n_k)$. Using this value of $\breve{z}(n_k)$ we obtain:

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h z(n_{k-1}). \tag{2.42}$$

In view of (2.40) 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

$$\begin{aligned} z(n) &= \Lambda_D{}^{n-n_k} z(n_k) \\ &= \Lambda_D{}^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h z(n_{k-1}) \\ &= \Lambda_D{}^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h z(n_{k-2}) \\ &\dots \\ &= \Lambda_D{}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \right)^k z_0. \end{aligned} \tag{2.43}$$

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

$$\begin{aligned} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} &= \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M^k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \\ &= \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix}. \end{aligned} \tag{2.44}$$

In view of (2.44) it is clear that the system response as in (2.39) may be written as:

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

♦

Note that the main difference between this theorem and the continuous version in Proposition 2.1 is in the state transition matrix used for the dynamics of the system in between updates ($\Lambda_D{}^{n-n_k}$ instead of $e^{\Lambda(t-t_k)}$). We now introduce an exponential global stability definition for the case of discrete plants [131].

Definition 2.5 The equilibrium $z = 0$ of a discrete time system described by $z(n+1) = f(n, z)$ with initial condition $z(n_0) = z_0$ is globally exponentially stable if there exists $\alpha > 0$ and $0 < \gamma < 1$ such that the solution

$$\|z(n)\| \leq \alpha\|z_0\|\gamma^n, \qquad \forall n \geq 0.$$

With this definition of stability we can now state the necessary and sufficient conditions for the exponential global stability of the system described by (2.38). Theorem 2.6 below for the discrete case is the corresponding result to Theorem 2.3 of the previous section for the continuous case. Again the norm used here is the 2-norm but any other consistent norm can also be used.

Theorem 2.6 *The system described by* (2.38) *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* $M_D = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are inside the unit circle.*

Proof Sufficiency. Taking the norm of the solution described as in Proposition 2.4:

$$\begin{aligned} \|z(n)\| &= \left\| {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \\ &\leq \|{\Lambda_D}^{n-n_k}\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\|. \end{aligned} \tag{2.45}$$

Now, the first term on the right-hand side of (2.45) satisfies:

$$\|{\Lambda_D}^{n-n_k}\| \leq (\overline{\sigma}(\Lambda_D))^{n-n_k} \leq (\overline{\sigma}(\Lambda_D))^h = K_1 \tag{2.46}$$

where $\overline{\sigma}(\Lambda_D)$ is the largest singular value of Λ_D. 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_D has eigenvalues with magnitude greater than one, $\|{\Lambda_D}^{n-n_k}\|$ can only grow a certain amount. This growth is completely independent of k.

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

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 \gamma_1^k \tag{2.47}$$

with $K_2 > 0,\ 0 < \gamma_1 < 1$.

Also note that k is a function of time so we can express the right term of (2.47) in terms of n:

$$K_2 \gamma_1^k = K_2 \gamma_1^{n/h} = K_2 \left(\gamma_1^{1/h} \right)^n = K_2 \gamma^n \tag{2.48}$$

where $0 < \gamma < 1$ since $h \geq 1$.

So from (2.45) using (2.46) and (2.48) we can conclude:

$$\|z(n)\| = \left\| \Lambda_D{}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_2 \gamma^n \cdot \|z_0\|. \tag{2.49}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the system is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ 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 0 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 $\Lambda_D{}^j$ has the following form:

$$\Lambda_D{}^j = \begin{bmatrix} W(j) & X(j) \\ Y(j) & Z(j) \end{bmatrix}. \tag{2.50}$$

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

$$\begin{aligned} &\Lambda_D{}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \\ &= \begin{bmatrix} W(n-n_k) & X(n-n_k) \\ Y(n-n_k) & Z(n-n_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(n-n_k)(W(h))^k & 0 \\ Y(n-n_k)(W(h))^k & 0 \end{bmatrix} z_0 \\ &\forall n \in \left[n_k, n_{k+1}\right). \end{aligned} \tag{2.51}$$

Now let us check the values of the solution at times n_{k+1}, that is the update time. We know that because of the update at this time the error is 0, and therefore:

$$z(n_{k+1}) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ 0 & 0 \end{bmatrix} z_0. \tag{2.52}$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D{}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ 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 0 for general initial condition x_0. That is,

$$\|x(n_{k+1})\| = \|\xi(k+1)\| = \left\|(W(h))^{k+1}x_0\right\| \to \infty \qquad \text{as} \qquad k \to \infty. \tag{2.53}$$

This clearly means that the system cannot be stable, and thus we have a contradiction. ♦

Example 2.4 Consider an example of the full state feedback setup using a discrete-time plant with parameters:

$$A = \begin{bmatrix} 0.89 & 1.23 \\ 0.08 & 0.98 \end{bmatrix}, \quad B = \begin{bmatrix} -0.04 \\ 1.19 \end{bmatrix} \tag{2.54}$$

The exact parameters of the plant are unknown. In general, a model containing nominal parameters of the real parameters is available. Assume that (2.54) is modeled using the following nominal parameters

$$\hat{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.55}$$

We will use the state feedback controller with control gain given by $K = [-0.12 \quad -0.7]$. We now search for the largest h such that $M_D = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has its eigenvalues inside the unit circle. Figure 2.10 shows the eigenvalue with maximum magnitude of this matrix. Recall that in the discrete-time case we only search for integer values of the update period equal or greater than 1. In this example we are able to use $h \leq 11$ in order to design a stable networked system using the model-based approach. Figure 2.11 shows the response for $h = 8$ and $h = 12$ and using the same initial conditions $x_0 = [-1 \ \ 0.2]^T$ and the model initial conditions are equal to 0. Linear interpolation is used when plotting the response of the systems. For each choice of h this figure shows the response of the plant and the model as well.

Example 2.5 Consider a discretized version of the instrument servo shown in Example 2.3. The nominal parameters are $\hat{\alpha} = 1$, $\hat{\beta} = 3$. The real parameters are given by $\alpha = -0.75$, $\beta = 2.58$. The discretization period is $T = 0.01$ s. The resulting discrete-time model parameters are used to design the controller: $K = [-30.1502 \ -5.8807]$. The same discrete-time model parameters are also used to implement the model in our discrete-time MB-NCS setup. Similar to the continuous-time implementation, stability is obtained for any integer value $h \geq 1$. The difference between implementing longer values of h is given by the transient response as it can be seen in Fig. 2.12 where the plant states converge in significantly longer time for $h = 50$ than for $h = 10$.

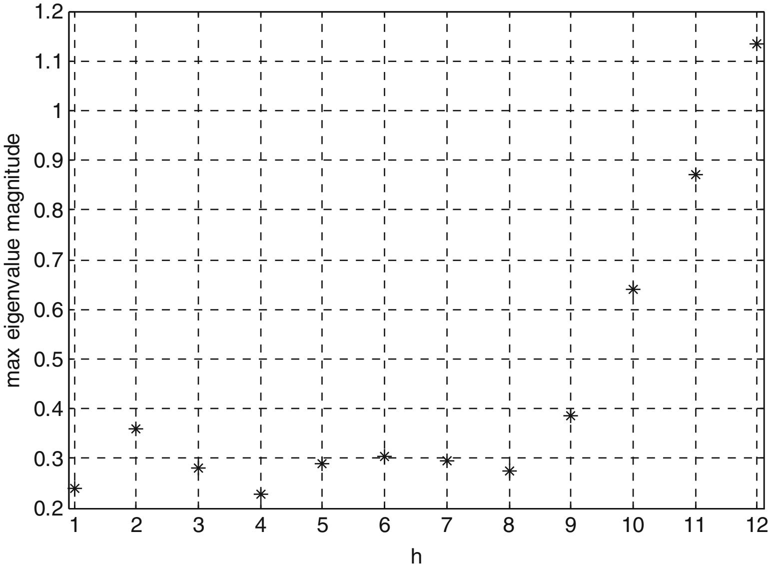

Fig. 2.10 Maximum eigenvalue magnitude of the test matrix M vs. the update time h

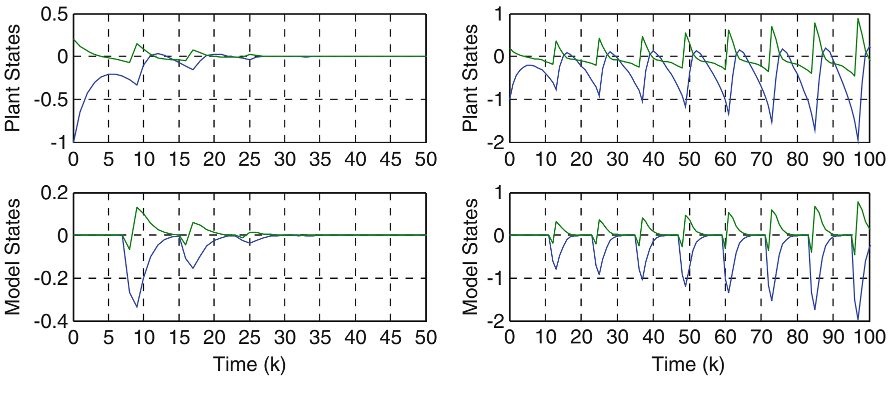

Fig. 2.11 Response of plant and model states. *Left*: $h = 8$. *Right*: $h = 12$

Remark Optimization problem. A similar optimization problem to the continuous-time case can also be considered in the discrete-time case. The problem is to find the optimal control input in the presence of model uncertainties and in the absence of feedback measurements for extended periods of time. Reducing network communication is also important when regulating the states of the system and minimizing the control effort.

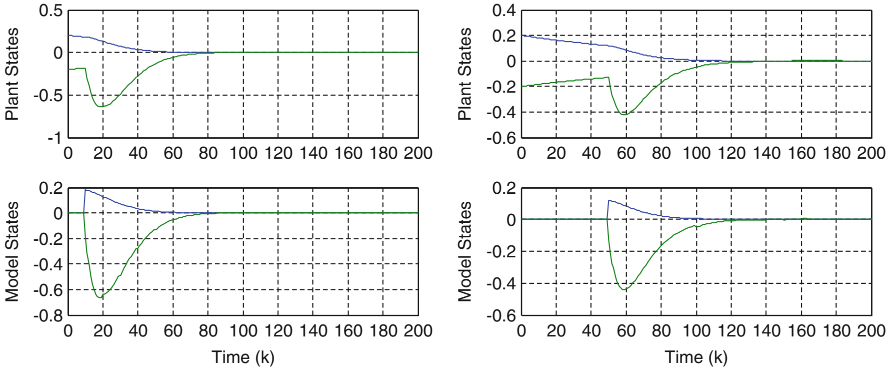

Fig. 2.12 Response of plant and model states. *Left*: $h = 10$. *Right*: $h = 50$

Mathematically, we try to solve the following problem:

$$\min_{u,\beta} x_T^T Q_T x_T + \sum_{n=0}^{T-1} x_n^T Q x_n + u_n^T R u_n + S\beta_n \tag{2.56}$$

where Q and Q_T are real, symmetric, and positive semi-definite matrices; R is a real, symmetric, and positive definite matrix; S is a positive weighting factor that penalizes network communication; x_T is the terminal state; and β_n is constrained to take on only two different values:

$$\beta_n = \begin{cases} 1 & measurement \;\; x_n \;\; is\, sent \\ 0 & measurement \;\; x_n \;\; is\, not\, sent \end{cases} \tag{2.57}$$

The optimal sequence of u_n and β_n, $n = 0,1,2,\ldots,T\text{-}1$, are of interest. This cost criterion can be applied to a discrete-time system or to a discretized version of continuous-time systems. The objective is to find the optimal control input and the optimal scheduling decisions (2.57). Solving this problem is the topic of Chap. 9.

2.4 Alternative Conditions for Stability of MB-NCS

In this section the stability of discrete-time MB-NCS is studied using a lifting approach. The stability of continuous-time systems is also considered using norms and a Lyapunov-based approach.

2.4.1 A Lifting Approach for Stability of Discrete-Time MB-NCS

In this section the traditional configuration for discrete-time MB-NCS will be studied using lifting techniques. In the lifting process, the input and output spaces are extended appropriately in order to obtain a Linear Time-Invariant (LTI) system description for sampled-data, multi-rate, or linear time-varying periodic systems. Since the lifted system is an LTI system, the available tools and results for LTI systems are applicable to the lifted system as well.

Applying the lifting approach to the traditional setup in MB-NCS, the stability problem is reformulated and necessary and sufficient conditions for discrete-time systems are derived which are the same conditions given in Sect. 2.3 derived using another approach. The main advantage of lifting here is that this strategy can be extended to the case when a communication network exists on both sides of the control loop, from sensor to controller and from controller to actuator. This topic will be covered in later chapters.

Lifting discrete-time signals and systems. This section provides a brief discussion on lifting discrete-time signals and systems based on [42]. Suppose there exist two periods h and h_s in a discrete-time setup and they are related by $h_s = h/r$, where r is some positive integer. For a discrete-time signal $v(k)$ referred to the sub-period h/r, that is, $v(0)$ occurs at time $t = 0$, $v(1)$ at $t = h/r$, $v(2)$ at $t = 2\ h/r$, and so on, the lifted signal $\underline{v}$ is defined as follows:

If $v = \{v(0), v(1), v(2), \ldots\}$, then

$$\underline{v} = \left\{ \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(r-1) \end{bmatrix}, \begin{bmatrix} v(r) \\ v(r+1) \\ \vdots \\ v(2r-1) \end{bmatrix}, \ldots \right\}. \tag{2.58}$$

The dimension of the lifted signal $\underline{v}(n)$ is r times the dimension of the original signal $v(n)$ and is regarded to the base period, i.e., $\underline{v}(n)$ occurs at time $t = nh$.

The lifting operator L is defined to be the map $v \rightarrow \underline{v}$. The inverse operator L^{-1} exists and is defined as follows:

If

$$\psi = \left\{ \begin{bmatrix} \psi_1(0) \\ \psi_2(0) \\ \vdots \\ \psi_n(0) \end{bmatrix}, \begin{bmatrix} \psi_1(1) \\ \psi_2(1) \\ \vdots \\ \psi_n(1) \end{bmatrix}, \ldots \right\} \tag{2.59}$$

and $v = L^{-1}\psi$, then $v = \{\psi_1(0), \psi_2(0), \ldots, \psi_n(0), \psi_1(1), \psi_2(1), \ldots, \psi_n(1), \ldots\}$.

A relevant feature of lifting is that it preserves inner products and norms. Let us see this for the case of l_2-norms. The norm of a signal $v \in l_2(\mathbb{Z}_+, \mathbb{R}^m)$ is given by:

$$\|v\|_2^2 = v(0)^T v(0) + v(1)^T v(1) + ... \tag{2.60}$$

and the norm of its lifted version $\underline{v} \in l_2(\mathbb{Z}_+, \mathbb{R}^{rm})$ is:

$$\begin{aligned} \|\underline{v}\|_2^2 &= \underline{v}(0)^T \underline{v}(0) + \underline{v}(1)^T \underline{v}(1) + ... \\ &= \begin{bmatrix} v(0) \\ : \\ v(r-1) \end{bmatrix}^T \begin{bmatrix} v(0) \\ : \\ v(r-1) \end{bmatrix} + \begin{bmatrix} v(r) \\ : \\ v(2r-1) \end{bmatrix}^T \begin{bmatrix} v(r) \\ : \\ v(2r-1) \end{bmatrix} + ... \\ &= v(0)^T v(0) + v(1)^T v(1) + ... \\ &= \|v\|_2^2. \end{aligned} \tag{2.61}$$

Now, let us consider a discrete-time finite dimensional LTI system G_d with underlying period h/r. Lifting the input and output signals so that the lifted signals correspond to the base period h results in the lifted system: $\underline{G}_d = LG_dL^{-1}$.

Assuming the state-space representation of the original system G_d is known and given by:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n). \end{aligned} \tag{2.62}$$

Then the state-space representation for the lifted system $\underline{G}_d$ is given by:

$$\begin{aligned} x((n+1)h) &= A^r x(nh) + \begin{bmatrix} A^{r-1}B & A^{r-2}B & \dots & B \end{bmatrix} \underline{u}(nh) \\ \underline{y}(nh) &= \begin{bmatrix} C \\ CA \\ : \\ CA^{r-1} \end{bmatrix} x(nh) + \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ : & & & \\ CA^{r-2}B & CA^{r-3}B & \dots & D \end{bmatrix} \underline{u}(nh). \end{aligned} \tag{2.63}$$

Lifting discrete-time MB-NCS. As it was described in Sect. 2.1, MB-NCS makes use of an explicit model of the plant which is added to the controller node to compute the control input based on the state of the model rather than on the plant state.

The dynamics of a discrete-time plant and model are given respectively by:

$$x(n+1) = Ax(n) + Bu(n) \tag{2.64}$$

$$\hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n) \tag{2.65}$$

where x is the state of the plant, $\hat{x}$ is the state of the model, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices A,B.

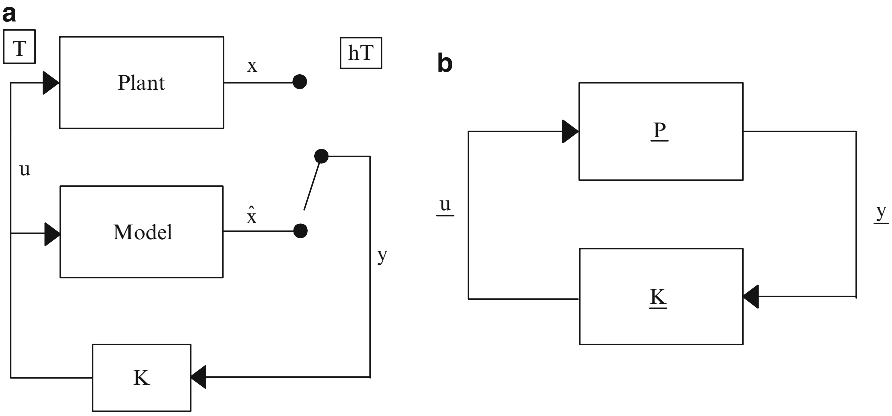

Fig. 2.13 Equivalent systems to MB-NCS. (**a**) Linear time-varying periodic system. (**b**) Lifted system

It will be advantageous to define the input u as:

$$u(n) = \begin{cases} Kx(n) & n = ih \\ K\hat{x}(n) & n = ih + j \end{cases} \tag{2.66}$$

for $i = 0, 1, 2....$ and $0 < j < h$ (j is also an integer), that is, the input is a function of the state of the plant at the time instant when we update the model, and a function of the state of the model otherwise.

A state feedback discrete-time MB-NCS can be seen as a linear time-varying periodic system as shown in (Fig. 2.13a) by considering an output y that is equal to $\hat{x}$ when the loop is open and equal to x when we have an update (closed loop). The system after applying lifting is represented in part b) of the same figure and is regarded as an LTI system with higher dimension input and output.

Note that the original period of the system is denoted by T and the period of the network by hT. Then, using the definition at the beginning of this section, we have that for this case $r = h$ since $hT/h = T$. The input $\underline{u}$ for the lifted system $\underline{P}$ and its output $\underline{y}$ are given by the following equations:

$$\underline{u}(nh) = \begin{bmatrix} u(nh) \\ u(nh+1) \\ : \\ u((nh+h-1) \end{bmatrix} = \begin{bmatrix} Kx(nh) \\ K\hat{x}(nh+1) \\ : \\ K\hat{x}((nh+h-1) \end{bmatrix} \tag{2.67}$$

$$\underline{y}(nh) = \begin{bmatrix} I \\ \hat{A} \\ \hat{A}^2 \\ \vdots \\ \hat{A}^{h-1} \end{bmatrix} x(nh) + \begin{bmatrix} 0 & 0 & \dots & 0 \\ \hat{B} & 0 & \dots & 0 \\ \hat{A}\hat{B} & \hat{B} & \dots & 0 \\ \vdots & & & \vdots \\ \hat{A}^{h-2}\hat{B} & \hat{A}^{h-3}\hat{B} & \dots & 0 \end{bmatrix} \underline{u}(nh). \tag{2.68}$$

The dimension of the state is preserved and the state equation expressed in terms of the lifted input is given by:

$$x((n+1)h) = A^h x(nh) + \begin{bmatrix} A^{h-1}B & A^{h-2}B & \dots & B \end{bmatrix} \underline{u}(nh). \tag{2.69}$$

The new controller is of the form: $\underline{K} = \begin{bmatrix} K & 0 & 0 & .. \\ 0 & K & 0 & \\ 0 & 0 & K & \\ \vdots & & & \end{bmatrix}$ where each zero block has the same dimensions as K.

Definition 2.7 [5] Asymptotic stability of discrete-time LTI system. The equilibrium $x = 0$ of a system described by $x(n+1) = Ax(n)$ is asymptotically stable if and only if all eigenvalues of A are within the unit circle of the complex plane (i.e., if $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of A, then $|\lambda_i| < 1,\ i = 1, \ldots, n$). In this case we say that the matrix A is Schur stable or simply, the matrix A is stable.

Theorem 2.8 *The lifted system is asymptotically stable if only if the eigenvalues of*

$$N = A^h + \sum_{j=0}^{h-1} A^{h-1-j} BK\left(\hat{A} + \hat{B}K\right)^j \tag{2.70}$$

lie strictly inside the unit circle.

Proof To prove this theorem, we note that (2.69) is the same as the state equation that characterizes the autonomous LTI system:

$$x((n+1)h) = \left(A^h + \sum_{j=0}^{h-1} A^{h-1-j} BK\left(\hat{A} + \hat{B}K\right)^j \right) x(nh). \tag{2.71}$$

Equation (2.71) can be obtained by directly substituting (2.67) in (2.69), and then substituting the value of each individual output by its equivalent in terms of the state *x(kh)*, i.e.,

$$\begin{aligned} \hat{x}(nh+1) &= \hat{A}x(nh) + \hat{B}u(nh) = \left(\hat{A} + \hat{B}K\right)x(nh) \\ \hat{x}(nh+2) &= \left(\hat{A} + \hat{B}K\right)^2 x(nh) \\ &\vdots \end{aligned} \tag{2.72}$$

The resulting equation can be simply expressed as (2.71) which characterizes a discrete-time LTI system of the form given in Definition 2.7, then the networked system is asymptotically stable if only if the eigenvalues of (2.70) lie inside the unit circle. ♦

2.4.2 Relation to Previous Results

The results presented in the previous section using a lifting approach are directly related to the stability conditions in Sect. 2.3. Since both results present necessary and sufficient conditions for stability of the discrete-time MB-NCS, we should be able to demonstrate that the conditions are the same. This is shown below.

From Theorem 2.6 a necessary and sufficient condition for stability of an MB-NCS with instantaneous feedback is given if the eigenvalues of

$$M_D(h) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \Lambda_D^h \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{2.73}$$

lie inside the unit circle, where

$$\Lambda_D = \begin{pmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{pmatrix}.$$

In order to establish the relation between the two theorems consider the transformation:

$$\overline{\Lambda} = P\Lambda_D P^{-1} = \begin{pmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{pmatrix} \tag{2.74}$$

where

$$P = \begin{pmatrix} I & 0 \\ I & -I \end{pmatrix}, \quad P^{-1} = \begin{pmatrix} I & 0 \\ I & -I \end{pmatrix} \tag{2.75}$$

and find the Z-transform of $M_D(n)$, $n = 0, 1, 2, \ldots$

$$Z\{M_D(n)\} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} Z\{\Lambda_D^n\} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

The Z-transform of Λ_D^n can be obtained according to:

$$Z\{\Lambda_D^n\} = (zI - \Lambda_D)^{-1} z = \left(zI - P^{-1}\overline{\Lambda}P\right)^{-1} z = P^{-1}\left(zI - \overline{\Lambda}\right)^{-1} Pz. \tag{2.76}$$

Substituting the last equation in the Z-transform of $M_D(n)$ we obtain:

$$Z\{M_D(k)\} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \left(zI - \overline{\Lambda}\right)^{-1} \begin{pmatrix} I & 0 \\ I & 0 \end{pmatrix} z$$
$$= \begin{pmatrix} (zI - A)^{-1} + (zI - A)^{-1}BK\left(zI - \left(\hat{A} + \hat{B}K\right)\right)^{-1} & 0 \\ 0 & 0 \end{pmatrix} z.$$

We now proceed to obtain the inverse Z-transform of the upper left sub-matrix which contains the eigenvalues of interest:

$$Z^{-1}\left\{(zI - A)^{-1}z + (zI - A)^{-1}BK\left(zI - \left(\hat{A} + \hat{B}K\right)\right)^{-1}z\right\} = A^n + \sum_{j=0}^{n-1} A^{n-1-j}BK\left(\hat{A} + \hat{B}K\right)^j. \tag{2.77}$$

Let $n = h$ in the above equation to obtain:

$$A^h + \sum_{j=0}^{h-1} A^{h-1-j}BK\left(\hat{A} + \hat{B}K\right)^j \tag{2.78}$$

which is exactly the same result obtained in Theorem 2.8, i.e., the eigenvalues of the first term of $M_D(h)$ correspond to the eigenvalues of (2.70). The other eigenvalues of $M_D(h)$ are not necessary in the analysis since they are always equal to 0 (they always lie inside the unit circle).

2.4.3 Additional Approaches for Stability of MB-NCS

We end this section by describing two additional approaches that provide conditions for stability of continuous-time MB-NCS that do not require exact knowledge of the plant parameters. These results can be used directly to compute the admissible range for h based only on certain bounds on the norms of the uncertainties. It is also important to clarify that the next conditions are sufficient only and, in general, very conservative, that is, the admissible update intervals h computed here are typically shorter than those obtained in Sect. 2.2.

Approach based on the norm of the state. In the first approach we set the requirement that the norm of the state of the plant should decrease at every sampling instant, that is, $\|x(t_{k+1})\| < \|x(t_k)\|$, for all $k = 0,1,\ldots$ and $h = t_{k+1} - t_k$. The plant and the model dynamics are given by (2.1) with $u = K\hat{x}$. The state error was defined in (2.2) and the error uncertainty matrices are expressed as follows.

$$\widetilde{A} = A - \hat{A}, \quad \widetilde{B} = B - \hat{B}. \tag{2.79}$$

By finding a control gain K that asymptotically stabilizes the model, i.e., $\hat{A} + \hat{B}K$ is Hurwitz, we can use the following bound:

$$\left\| e^{\left(\hat{A} + \hat{B}K\right)t} \right\| \leq \alpha e^{-\beta t}, \qquad \alpha, \beta > 0. \tag{2.80}$$

Theorem 2.9 *Assume that the pair* $\left(\hat{A}, \hat{B}\right)$ *is stabilizable. The model-based networked system described by* (2.1) *is asymptotically stable if:*

$$1 - \alpha\left(e^{-\beta h} + \frac{\widetilde{K}}{\beta + \beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right)\right) > 0 \tag{2.81}$$

where $\|A\| \leq \left\|\hat{A}\right\| + \left\|\widetilde{A}\right\| \leq \beta_2$, $\left\|\widetilde{A} + \widetilde{B}K\right\| \leq \widetilde{K}$, $h = t_{k+1} - t_k$.

Proof From (2.2) we can obtain the next expression for the state error:

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ae + \left(\widetilde{A} + \widetilde{B}K\right)\hat{x}. \tag{2.82}$$

The response of the state error at any given time as a function of the last received measurement is:

$$\begin{aligned} e(t) &= e^{A(t-t_k)} e\left(t_k\right) - \int_{t_k}^{t} e^{A(t-\tau)} \left(\widetilde{A} + \widetilde{B}K\right) \hat{x}\left(\tau - t_k\right) d\tau \\ &= -\int_{t_k}^{t} e^{A(t-\tau)} \left(\widetilde{A} + \widetilde{B}K\right) e^{\left(\hat{A} + \hat{B}K\right)(\tau - t_k)} d\tau \cdot x(t_k). \end{aligned} \tag{2.83}$$

then, a bound for the norm of the state error is:

$$\|e(t)\| \leq \frac{\alpha\widetilde{K}\|x(t_k)\|}{\beta + \beta_2}\left(e^{\beta_2(t-t_k)} - e^{-\beta(t-t_k)}\right). \tag{2.84}$$

It is also clear from (2.2) that the norm of the state satisfies:

$$\|x(t)\| < \|\hat{x}(t)\| + \|e(t)\|. \tag{2.85}$$

Substituting (2.84) in (2.85) we have, using again the bound (2.80):

$$\|x(t)\| \leq \alpha e^{-\beta(t-t_k)}\|x(t_k)\| + \frac{\alpha\widetilde{K}\|x(t_k)\|}{\beta + \beta_2}\left(e^{\beta_2(t-t_k)} - e^{-\beta(t-t_k)}\right). \tag{2.86}$$

In order to satisfy the initial condition on the norm of the state we require that:

$$\|x(t_k)\| - \|x(t_k)\|\left(\alpha e^{-\beta h} + \frac{\alpha\widetilde{K}}{\beta+\beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right)\right) > 0 \tag{2.87}$$

which is equivalent to (2.81). ♦

Approach based on Lyapunov functions. In the second approach we set the requirement that the discrete-time Lyapunov function along the trajectories of the system decreases at every sampling instant, i.e., $V(x(t_{k+1})) - V(x(t_k)) < 0$, for $k = 0,1,\ldots$

Theorem 2.10 *Assume that the pair* $(\hat{A}, \hat{B})$ *is stabilizable. The model-based networked system described by* (2.1) *is asymptotically stable if*:

$$2\overline{\sigma}(F)G_h + G_h^2 < \frac{q}{\overline{\sigma}(P)} \tag{2.88}$$

where $F = e^{(\hat{A}+\hat{B}K)h}$, $G_h = \frac{\alpha\widetilde{K}}{\beta+\beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right) \geq \overline{\sigma}(\Delta(h))$, $\Delta(h) = \int_0^h e^{A(h-\tau)}$ $\left(\widetilde{A}+\widetilde{B}K\right)e^{(\hat{A}+\hat{B}K)\tau}d\tau$, $\|A\| \leq \|\hat{A}\| + \|\widetilde{A}\| \leq \beta_2$, α and β are given by (2.80).

Proof Using (2.2) and (2.83), the state of the plant can be expressed by:

$$\begin{aligned} x(t_{k+1}) &= \left(e^{(\hat{A}+\hat{B}K)h} + \int_0^h e^{A(h-\tau)}\left(\widetilde{A}+\widetilde{B}K\right)e^{(\hat{A}+\hat{B}K)\tau}d\tau\right)\cdot x(t_k) \\ &= (F+\Delta(h))x(t_k). \end{aligned} \tag{2.89}$$

Next we define a quadratic Lyapunov function $V = x^T Px$, that is evaluated at the update instants as follows:

$$V(x(t_{k+1})) - V(x(t_k)) = x(t_k)^T\left[(F+\Delta(h))^T P(F+\Delta(h)) - P\right]x(t_k). \tag{2.90}$$

Note that a stabilizing controller for the model can be found. F is a stable discrete-time matrix that satisfies:

$$F^T PF - P = -Q \tag{2.91}$$

for symmetric, positive definite matrices P and Q. Then a sufficient condition for stability of the system is given in terms of the perturbation $\Delta(h)$, as follows

$$\begin{aligned} &(F+\Delta(h))^T P\big(F+\Delta(h)\big) - P \\ &\quad = -Q + \Delta(h)^T PF + F^T P\Delta\big(h\big) + \Delta\big(h\big)^T P\Delta\big(h\big) \\ &\quad \leq -\underline{\sigma}(Q) + \overline{\sigma}\big[\Delta\big(h\big)^T PF + F^T P\Delta\big(h\big) + \Delta\big(h\big)^T P\Delta\big(h\big)\big] \\ &\quad \leq -q + 2\overline{\sigma}(PF)G_h + \overline{\sigma}\big(P\big)G_h^2 < 0 \end{aligned} \tag{2.92}$$

where $q = \underline{\sigma}(Q)$ represents the smallest singular value of Q and $\overline{\sigma}(P)$ represents the largest singular value of P. If (2.88) holds, then (2.92) is satisfied. ♦

Recall that the application of the results in Sect. 2.2 to real world problems involves a repeated computation of eigenvalues for different values of parameters uncertainties as explained in Example 2.2. The results in this subsection can be used in a direct way based on a priori information about uncertainty bounds. The disadvantage is that these conditions are sufficient only and may be conservative in general, resulting in smaller update intervals h.

2.5 Notes and References

In this chapter we introduced the basic MB-NCS architecture that will be used throughout this book. This framework uses the available knowledge of the real plant dynamics encapsulated in the plant model to perform an open-loop estimation of the real plant state that is used to compute the control input for the intervals of time that the controller does not receive any measured feedback information. The main results of this chapter were presented in Theorem 2.3 and Theorem 2.6. Theorem 2.3 provides necessary and sufficient conditions for exponential stability of the networked system as a function of the update intervals which are assumed to be periodic. Theorem 2.6 provides the corresponding conditions for discrete-time systems. In both cases, it is assumed that the sensor is able to measure the entire state of the system. Alternative stability conditions were given in Sect. 2.4. In subsequent chapters, different extensions, alternative sampling methodologies, and architectures will be explored. Additionally, different common problems in control theory will be discussed from the model-based networked perspective.

The MB-NCS architecture shown in Sect. 2.1 was first proposed by Montestruque and Antsaklis [186, 187] and the results shown in Sects. 2.2 and 2.3 are primarily based on work published in those two references and also in [188]. The lifting approach for MB-NCS discussed in Sect. 2.4 was first published by Garcia and Antsaklis, [82]. The lifting methods used in that section are based on common lifting techniques presented in [42]. See also [16, 26, 130] for additional discussion of lifting techniques.

The alternative stability conditions for continuous-time systems of Sect. 2.4 represent original work presented in this book and is an adaptation of the work by Montestruque [185] on nonlinear systems.

The MB-NCS framework has been followed by different authors to consider specific problems in NCS. Orihuela et al. [207] have derived conditions for stability of MB-NCS with parametric uncertainties based on the theory of interval matrices. The authors of [197] compare different control decisions for a nonlinear NCS subject to sensor losses. The options include zero control, last available control, and open-loop control; the last one corresponds to using the nominal nonlinear model of the plant. It is proved in that paper that under certain conditions the use of the model exceeds in performance the other two choices to decide the control input in the absence of feedback information. The MB-NCS structure is used by Yu et al. [283] to study singularly perturbed systems where the sensor is connected to the controller/actuator node using a communication network. Singularly perturbed systems refer to the two-time scale systems that appear due to the presence of small parasitic parameters multiplying the time derivatives of some states of the system. It is difficult, in general, to control this type of systems since the controller has to react simultaneously to both the slow and fast modes of the system. Mu et al. [196] provided a different analysis of MB-NCS with output feedback and constant network delays where the model is updated using directly the received output $y(t_k) = Cx(t_k)$.

Recent work has been produced independently and it shares many characteristics of the MB-NCS framework, indicating the fundamental advantages of using the model dynamics when operating and controlling dynamical systems in the absence of continuous feedback information. Motivated by human operations, the authors of [80] and [126] point out that in general a human operator scans information intermittently and operates continuously the controlled system; the intermittent characteristic in this case refers to the same situation presented in this chapter, that is, a single measurement is used to update the internal model and generate the control input. For a skillful operator, the information is scanned less frequently. Between update intervals the control input is generated the same way as it was shown in the MB-NCS framework, that is, an imperfect model of the system is used to generate an estimate of the state and periodic measurements are used to update the state of this model. In the output feedback case a stochastic estimator is implemented with the assumption that the statistical properties of the measurement noise are known. In both cases the authors provide conditions for stability based on the length of the sampling interval.

In the networked framework shown in [140–142, 165], the model is assumed to match the dynamics of the system exactly; however, the system is subject to unknown input disturbances. The main idea of the approach in those papers is the same as the one described in this chapter, that is, to use the nominal model to generate estimates of the current state of the system. Since the system is subject to unknown disturbances and the model is executed with zero input disturbance, then a difference between the states is expected and the sensor updates transmitted over a digital communication network are used to reset this difference between the states of the plant and of the model at the update time instants.

There exist similarities and significant differences between the MB-NCS approach and the developed and mature Model Predictive Control (MPC) approach. MPC also uses model of the system for control; specifically, the model is used to predict the future output behavior; a tracking error is defined using this prediction and the desired reference and the control action are computed online. The purpose of the control action is to drive the state of the system to a reference position in an optimal fashion while satisfying the existing constraints. MPC relies in frequent feedback measurements in order to update the predicted control sequences. Meanwhile, in MB-NCS the main constraint is imposed in the form of reducing measurement update rates. Recent work related to NCS [24, 255, 288] has been reported where model predictive controllers have been successfully applied to deal with the usual input and output constraints, but also consider the bandwidth limitations of the network. The additional constraint aims to reduce the traffic in the communication network by using the explicit model to generate the appropriate input in the absence of continuous feedback.

http://www.springer.com/978-3-319-07802-1

Model-Based Control of Networked Systems
Garcia, E.; Antsaklis, P.; Montestruque, L.A.
2014, XVI, 382 p. 127 illus., 85 illus. in color., Hardcover
ISBN: 978-3-319-07802-1
A product of Birkhäuser Basel