# Chapter 2
# Fundamentals

In analysis and synthesis of NCSs and WNCSs, various approaches have been proposed, such as stochastic systems, Markovian jump systems, switched systems and time-delay systems. To help readers understand the book well, some fundamentals on stability analysis, controller and filter design are first presented for linear time-invariant (LTI) systems, Markovian jump systems and switched systems. Several lemmas are introduced, and some of them will be used in this chapter. The proof of these lemmas can be found in the literature and thus are omitted in this book.

## 2.1 Mathematical Preliminaries

Some basic mathematical preliminaries relevant to this book are given in this section.

**Lemma 2.1** ([1]) *For a given matrix* $S = \begin{bmatrix} S_{11} & S_{12} \\ * & S_{22} \end{bmatrix}$*, the following three statements are equivalent:*

*(1)* $S < 0$*;*
*(2)* $S_{11} < 0,\ S_{22} - S_{12}^T S_{11}^{-1} S_{12} < 0$*;*
*(3)* $S_{22} < 0,\ S_{11} - S_{12} S_{22}^{-1} S_{12}^T < 0$.

**Lemma 2.2** ([2]) *For given matrices* $K_1$*,* $K_2$ *and* $K_3$ *with appropriate dimensions, and* $K_1$ *satisfying* $K_1 = K_1^T$*, then there holds*

$$K_1 + K_2 \Delta(k) K_3^T + K_3 \Delta^T(k) K_2^T < 0 \tag{2.1}$$

*for all* $\Delta^T(k)\Delta(k) \le I$ *if and only if there exists a scalar* $\varepsilon > 0$ *such that*

$$K_1 + \varepsilon K_2 K_2^T + \varepsilon^{-1} K_3 K_3^T < 0. \tag{2.2}$$



D. Zhang et al., *Filtering and Control of Wireless Networked Systems*,
Studies in Systems, Decision and Control 97, DOI 10.1007/978-3-319-53123-6_2

**Lemma 2.3** ([3]) *For matrices $A$, $Q = Q^T$ and $P > 0$, the following matrix inequality,*

$$A^T P A - Q < 0, \tag{2.3}$$

*holds if and only if there exists a matrix $W$ of appropriate dimensions such that*

$$\begin{bmatrix} -Q & A^T W \\ * & P - W - W^T \end{bmatrix} < 0. \tag{2.4}$$

## 2.2 LTI Systems

A dynamic system is usually modeled as a differential or difference equation. A continuous-time linear time-invariant system is usually described by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.5}$$

and a discrete-time system is

$$x(k+1) = Ax(k) + Bu(k). \tag{2.6}$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the state vector. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are two constant matrices. It follows from the Lyapunov stability theory that the above systems are asymptotically sable if the state $x(t)$ or $x(k)$ tends to its equilibrium point when the time goes to infinity. The above system is exponentially stable if the state $x(t)$ or $x(k)$ tends to its equilibrium point with an exponential decay rate. Taking the discrete-time system (2.6) as an illustration, it is said to be exponentially stable if there exist some scalars $\delta > 0$ and $0 < \beta < 1$, such that the state of (2.6) satisfies $\|x(k)\| < \delta\beta^{k-k_0} \|x(k_0)\|, \forall k \geq k_0$.

There are various ways to check whether an LTI system is stable or not. We focus on the Lyapunov stability theory and linear matrix inequality based conditions. We first discuss the scenario when system (2.5) and (2.6) are in the absence of input, i.e., $u = 0$.

**Proposition 2.1** *The continuous-time LTI system (2.5) with u=0 is said to be asymptotically sable if there exists a positive definite matrix $P > 0$ such that the following inequality is true:*

$$A^T P + PA < 0. \tag{2.7}$$

*Proof* Consider the Lyapunov function candidate $V(x(t)) = x^T(t)Px(t)$. The derivative of this Lyapunov function is

$$
\begin{aligned}
\dot{V}(x(t)) &= \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t) \\
&= (Ax(t))^T Px(t) + x^T(t)P\,(Ax(t)) \\
&= x^T(t)\left(A^T P + PA\right)x(t).
\end{aligned}
\tag{2.8}
$$

One can see that $\dot{V}(x(t)) < 0$ if (2.7) holds. It follows from the Lyapunov stability theory, the system (2.5) with $u = 0$ is asymptotically stable.

**Proposition 2.2** *The discrete-time LTI system (2.6) with $u = 0$ is said to be asymptotically sable if there exists a positive definite matrix $P > 0$ such that the following inequality is true:*

$$
A^T PA - P < 0. \tag{2.9}
$$

*Proof* Use the Lyapunov function $V(x(k)) = x^T(k)Px(k)$. The difference of Lyapunov function is

$$
\begin{aligned}
\Delta V(x(k)) &= V(x(k+1)) - V(x(k)) \\
&= x^T(k+1)Px(k+1) - x^T(k)Px(k) \\
&= (Ax(k))^T P\,(Ax(k)) - x^T(k)Px(k) \\
&= x^T(k)\left(A^T PA - P\right)x(k).
\end{aligned}
\tag{2.10}
$$

One can see that $\Delta V(x(k)) < 0$ if (2.9) holds. It follows from the Lyapunov stability theory, the system (2.6) with $u = 0$ is asymptotically stable.

For the control system (2.5) and (2.6), the state feedback controller is a simple and effective method to adjust the dynamic of systems. The overall control system is given by

$$
\begin{cases}
\dot{x}(t) = Ax(t) + Bu(t), \\
u(t) = -Kx(t),
\end{cases}
\tag{2.11}
$$

where $K \in \mathbb{R}^{m\times n}$ is the controller gain to be determined.

**Proposition 2.3** *The closed-loop system (2.11) is asymptotically stable if there exist positive definite matrix $X$ and an appropriate matrix $Y$ such that the following inequalities*

$$
\begin{cases}
AX + XA^T - Y^T B^T - BY < 0, \\
X > 0,
\end{cases}
\tag{2.12}
$$

*hold. Moreover, the controller gain is determined by $K = YX^{-1}$.*

*Proof* The closed-loop system can be described as $\dot{x}(t) = (A - BK)x(t)$. By replacing $A$ in (2.7) by $A - BK$, it is easy to see that the closed-loop system is stable when the following inequality,

$$(A - BK)^T P + P(A - BK) < 0, \tag{2.13}$$

holds. Since $P = P^T > 0$, then $P$ is invertible. Left- and right- multiplying the above inequality by $P^{-1}$ yields

$$AP^{-1} + P^{-1}A^T - \left(P^{-1}K^T\right)B^T - B\left(KP^{-1}\right) < 0, \tag{2.14}$$

which is (2.12) by assigning $X = P^{-1}$ and $Y = KP^{-1}$.

The following proposition deals with stabilization of a discrete-time system:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ u(k) = -Kx(k). \end{cases} \tag{2.15}$$

**Proposition 2.4** *The closed-loop system (2.15) is asymptotic stable if there exist positive definite matrix $X$ and an appropriate matrix $Y$ such that the following inequality,*

$$\begin{bmatrix} -X & XA^T - YB^T \\ * & -X \end{bmatrix} < 0, \tag{2.16}$$

*holds. Moreover, the controller gain is determined by $K = YX^{-1}$.*

*Proof* The closed-loop system can be described as $x(k+1) = (A - BK)x(k)$. By replacing $A$ in (2.9) by $A - BK$, it is easy to see that the closed-loop system is stable when the following inequality holds:

$$(A - BK)^T P(A - BK) < 0. \tag{2.17}$$

By using Lemma 2.1, (2.17) is equivalent to

$$\begin{bmatrix} -P & (A - BK)^T P \\ * & -P \end{bmatrix} < 0. \tag{2.18}$$

Since $P = P^T > 0$, then $P$ is invertible. Left- and right- multiplying the above inequality by $\mathrm{diag}\{P^{-1}, P^{-1}\}$ gives

$$\begin{bmatrix} -P^{-1} & P^{-1}(A - BK)^T \\ * & -P^{-1} \end{bmatrix} < 0. \tag{2.19}$$

which is (2.16) by assigning $X = P^{-1}$ and $Y^T = P^{-1}K^T$.

## 2.3 Markovian Jump Systems

A useful category of system models is those in which the system operates in multiple modes. The switching between these modes introduces non-linearity into the overall system description even though each individual mode is linear. A general theory of such systems is presented by the hybrid systems community. However, much tighter results can be developed if some further assumptions are made, for example the mode switches are governed by a stochastic process that is statistically independent from the state values. In the case when the stochastic process can be described by a Markov chain, the system is called a Markovian jump linear system. The Markovian jump system has been applied to the modeling of networked systems, and we focus on the discrete-time system.

Consider a discrete-time Markovian jump system:

$$x(k+1) = A_{r(k)}x(k), \tag{2.20}$$

where $x(k)$ is the state vector, $r(k) \in \Theta = \{1, 2, \ldots, m\}$ is the switching law and the transition probability is denoted as $\Pr ob\,(r_{k+1} = j|\, r_k = i) = q_{ij}$, and $0 \leq q_{ij} \leq 1$, which constitutes the transition matrix $Q$. Such a system has been studied for a long time in the fault isolation community, and received new impetus with the adventure of networked control systems. For example, as mentioned in Chap. 1, the packet dropouts process can be modeled as a two state Markovian jump system. The stochastic access constraint phenomena can also be modeled as a Markovian jump system. Then, it is necessary to introduce some basic knowledge of this system.

Since the Markovian jump linear system is a stochastically varying system, numerous notions of stability may be defined. We will primarily be interested in mean-square stability, that is the state of a Markovian system tends to its equilibrium point in the mean-square sense when the time goes to infinity. Mathematically, $\mathbb{E}\left\{\sum\limits_{k=0}^{\infty} \|x(k)\|^2 |\chi(0)\right\} < \infty$, where $\chi(0)$ is the initial condition.

A sufficient stability condition for the Markovian jump system (2.20) is given as follows.

**Proposition 2.5** *The discrete-time Markovian jump system (2.20) is said to be mean-square sable if there exist positive definite matrices $P_i > 0$, such that the following inequalities are all true:*

$$A_i^T \left(\sum_{j=1}^{m} q_{ij} P_j\right) A_i - P_i < 0, \quad i, j \in \Theta. \tag{2.21}$$

*Proof* Let $V(x(k)) = x^T(k)P_{r(k)}x(k)$, $r(k) = i$ and $r(k+1) = j$, it follows from that

$$\begin{aligned}\mathbb{E}\{\Delta V(x(k))\} &= \mathbb{E}\{V(x(k+1)) - V(x(k))\}\\ &= \mathbb{E}\left\{x^T(k+1)\left(\sum_{j=1}^{m} q_{ij}P_j\right)x(k+1) - x^T(k)P_i x(k)\right\}\\ &= \mathbb{E}\left\{(A_i x(k))^T\left(\sum_{j=1}^{m} q_{ij}P_j\right)(A_i x(k)) - x^T(k)P_i x(k)\right\}\\ &= \mathbb{E}\left\{x^T(k)\left[A_i^T\left(\sum_{j=1}^{m} q_{ij}P_j\right)A_i - P_i\right]x(k)\right\}.\end{aligned} \tag{2.22}$$

One can see that $\mathbb{E}\{\Delta V(x(k))\} < 0$ if (2.21) holds. It follows from the Lyapunov stability theory, the system (2.20) is asymptotically stable in the mean-square sense.

Proposition 2.5 gives a sufficient condition for the asymptotically stability of system (2.20). We now consider a system with the input, and the overall system is described by

$$\begin{cases} x(k+1) = A_{r(k)}x(k) + B_{r(k)}u(k),\\ u(k) = K_{r(k)}x(k), \end{cases} \tag{2.23}$$

where $u(k)$ is the control input and $K_{r(k)}$ is the feedback gain, which is to be determined. The following algorithm can be used to determine the controller gain.

**Proposition 2.6** *For the discrete-time Markovian jump system (2.23), it is mean-square stable if there exist positive-definite matrices $Q_i > 0$ and matrices $\bar{K}_i$, $i = 1, 2, \ldots, m$, such that the following inequalities,*

$$\begin{bmatrix} -Q_i & \sqrt{q_{i1}}\left(Q_1 A_i^T + \bar{K}_i^T B_i^T\right) & \cdots & \sqrt{q_{im}}\left(Q_m A_i^T + \bar{K}_i^T B_i^T\right)\\ * & -Q_1 & \cdots & 0\\ \vdots & * & \ddots & \vdots\\ * & * & \cdots & -Q_m \end{bmatrix} < 0, \tag{2.24}$$

*hold. Moreover, the controller gain can be determined by $K_i = \bar{K}_i Q_i^{-1}$.*

*Proof* By replacing $A_i$ in (2.21) by $A_i + B_i K_i$, the system (2.23) is stable if

$$(A_i + B_i K_i)^T\left(\sum_{j=1}^{m} q_{ij}P_j\right)(A_i + B_i K_i) - P_i < 0, \quad i, j \in \Theta, \tag{2.25}$$

which is equivalent to

$$\begin{bmatrix} -P_i & \sqrt{q_{i1}}\left(A_i^T + K_i^T B_i^T\right)P_1 & \cdots & \sqrt{q_{im}}\left(A_i^T + K_i^T B_i^T\right)P_m\\ * & -P_1 & \cdots & 0\\ \vdots & * & \ddots & \vdots\\ * & * & \cdots & -P_m \end{bmatrix} < 0. \tag{2.26}$$

By left- and right- multiplying $\mathrm{diag}\{P_i^{-1}, P_1^{-1}, \ldots, P_m^{-1}\}$ and its transpose to (2.26), respectively, we have

$$\begin{bmatrix} -P_i^{-1} & \sqrt{q_{i1}}P_1^{-1}\left(A_i^T + K_i^T B_i^T\right) & \cdots & \sqrt{q_{im}}P_m^{-1}\left(A_i^T + K_i^T B_i^T\right) \\ * & -P_1^{-1} & \cdots & 0 \\ \vdots & * & \ddots & \vdots \\ * & * & \cdots & -P_m^{-1} \end{bmatrix} < 0. \tag{2.27}$$

Let $P_i^{-1} = Q_i$ and $\bar{K}_i^T = Q_i K_i^T$. We see that (2.27) is the same as (2.24).

Apart from stability and control problems, the state estimation problem, also called as the filtering is another important research topic. The main purpose is to estimate the plant state by using the available measurement signals. The estimated state can be used for state monitoring when the state is not measurable. It can also be used for controller design when full state information is not available. In the last decades, many filtering approaches have been proposed such as Kalman filtering, $H_2$ filtering, $H_\infty$ filtering and $l_2$-$l_\infty$. It is well known that the standard Kalman filter is sensitive to modeling errors and the distribution of noise should be Gaussian white noise. In practice, the modeling error is inevitable and one may not always have the distribution of noise. We focus on the $H_\infty$ filtering approach in this book as it requires less information than the Kalman filtering approach. To start, we consider the following system:

$$\begin{cases} x(k+1) = A_i x(k) + B_i w(k), \\ z(k) = L_i x(k), \end{cases} \tag{2.28}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $w(k) \in \mathbb{R}^m$ is the unknown disturbance and usually assumed to belong to $l_2[0, \infty)$ in the $H_\infty$ filtering framework. $z(k) \in \mathbb{R}^p$ is the signal to be estimated, which can be a partial state vector. The measurement signal $y(k)$ is usually described by

$$y(k) = C_i x(k) + D_i w(k). \tag{2.29}$$

To estimate $z(k)$ in (2.28), one uses the following filter:

$$\begin{cases} \hat{x}(k+1) = A_{fi}\hat{x}(k) + B_{fi} y(k), \\ z_f(k) = C_{fi}\hat{x}(k), \end{cases} \tag{2.30}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the state of the filter, and $z_f \in \mathbb{R}^p$ is the estimation of $z(k)$ in (2.28). $A_{fi}$, $B_{fi}$ and $C_{fi}$ are the filter gains to be determined. Based on (2.28)–(2.30), we have the filtering error system described by

$$\begin{cases} \eta(k+1) = \tilde{A}_i \eta(k) + \tilde{B}_i w(k), \\ e(k) = \tilde{C}_i \eta(k), \end{cases} \tag{2.31}$$

where

$$\eta(k) = \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix}, e(k) = z(k) - z_f(k),$$

$$\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ B_{fi}C_i & A_{fi} \end{bmatrix}, \tilde{B}_i = \begin{bmatrix} B_i \\ B_{fi}D_i \end{bmatrix}, \tilde{C}_i = \begin{bmatrix} L_i & -C_{fi} \end{bmatrix}.$$

The purpose of the estimation problem is to design the filter in the form of (2.30) such that the filtering error system (2.31) is mean-square stable and achieves a prescribed $H_\infty$ filtering performance level. That is

- system (2.31) is stochastically stable with $w(k) = 0$;
- under the zero initial conditions, $\mathbb{E}\left\{\sum_{s=0}^{t}\left[e^T(s)e(s)\right]\right\} < \gamma^2\sum_{s=0}^{t}\left[w^T(s)w(s)\right]$ holds.

The following proposition gives a sufficient condition for the existence of such a filter.

**Proposition 2.7** *The filtering error system (2.31) is asymptotically stable in the mean-square sense and with a prescribed $H_\infty$ performance level $\gamma$, if there exist positive-definite matrices $P_i$ such that the following inequalities are true,*

$$\begin{bmatrix} -P_i & 0 & \tilde{A}_i^T\bar{P}_i & \tilde{C}_i^T \\ * & -\gamma^2 I & \tilde{B}_i^T\bar{P}_i & 0 \\ * & * & -\bar{P}_i & 0 \\ * & * & * & -I \end{bmatrix} < 0, \tag{2.32}$$

*where* $\bar{P}_i = \sum_{j=1}^{m} q_{ij}P_j$.

*Proof* Let $V(x(k)) = x^T(k)P_{r(k)}x(k)$, $r(k) = i$ and $r(k+1) = j$, then

$$\begin{aligned} &\mathbb{E}\left\{V(x(k+1)) - V(x(k)) + e^T(k)e(k) - \gamma^2 w(k)w(k)\right\} \\ &\quad = \mathbb{E}\left\{\eta^T(k+1)\bar{P}_i\eta(k+1) - \eta^T(k)P_i\eta(k)\right\} \\ &\qquad + \mathbb{E}\left\{\left(\tilde{C}_i x(k)\right)^T\left(\tilde{C}_i x(k)\right) - \gamma^2 w(k)w(k)\right\} \\ &\quad = \mathbb{E}\left\{\left[\tilde{A}_i\eta(k) + \tilde{B}_i w(k)\right]^T\bar{P}_i\left[\tilde{A}_i\eta(k) + \tilde{B}_i w(k)\right]\right\} \\ &\qquad - \eta^T(k)P_i\eta(k) \\ &\quad + \ \mathbb{E}\left\{\left(\tilde{C}_i x(k)\right)^T\left(\tilde{C}_i x(k)\right) - \gamma^2 w(k)w(k)\right\} \\ &\quad = \mathbb{E}\left\{\bar{\eta}^T(k)\left(\Omega + \Omega_1\bar{P}_i\Omega_1^T + \Omega_2\Omega_2^T\right)\bar{\eta}(k)\right\}, \end{aligned} \tag{2.33}$$

where

$$\bar{\eta}(k) = \begin{bmatrix} \eta(k) \\ w(k) \end{bmatrix}, \Omega = \begin{bmatrix} -P_i & 0 \\ * & -\gamma^2 I \end{bmatrix},$$
$$\Omega_1 = \begin{bmatrix} \tilde{A}_i^T \\ \tilde{B}_i^T \end{bmatrix}, \Omega_2 = \begin{bmatrix} \tilde{C}_i^T \\ 0 \end{bmatrix}.$$

By Lemma 2.1, it is easy to see that (2.32) guarantees $\Omega + \Omega_1 \bar{P}_i \Omega_1^T + \Omega_2 \Omega_2^T < 0$. Then, $\mathbb{E}\left\{V(x(k+1)) - V(x(k)) + e^T(k)e(k) - \gamma^2 w(k)w(k)\right\}$. By summing both sides of this inequality, we have

$$\mathbb{E}\left\{\sum_{s=0}^{t}\left[V(x(s+1)) - V(x(s)) + e^T(s)e(s) - \gamma^2 w(s)w(s)\right]\right\} < 0. \tag{2.34}$$

It follows from $V(x(0)) = 0$ and $V(x(s+1)) \geq 0$ that

$$\mathbb{E}\left\{\sum_{s=0}^{t}\left[e^T(s)e(s) - \gamma^2 w(s)w(s)\right]\right\} < 0, \tag{2.35}$$

that is

$$\mathbb{E}\left\{\sum_{s=0}^{t}\left[e^T(s)e(s)\right]\right\} < \gamma^2 \sum_{s=0}^{t}[w(s)w(s)]. \tag{2.36}$$

We can conclude that system (2.31) is mean-square stable and also has a prescribed $H_\infty$ performance $\gamma$.

With the help of Proposition 2.7, we can determine the filter gain parameters by using the following proposition.

**Proposition 2.8** *The $H_\infty$ filtering problem is solvable if there exist positive definite matrices $P_i$ and some matrices $W_i$ with appropriate dimensions such that the following inequalities,*

$$\begin{bmatrix} -P_i & 0 & \Xi_1 & \Xi_3 \\ * & -\gamma^2 I & \Xi_2 & 0 \\ * & * & \tilde{P}_i & 0 \\ * & * & * & -I \end{bmatrix} < 0, \tag{2.37}$$

*hold for all $i \in \Theta$. Then the filter gains can be determined by $A_{fi} = W_{3i}^{-T} A_{Fi}$, $B_{fi} = W_{3i}^{-T} B_{Fi}$, and $C_{fi} = C_{Fi}$, where*

$$\Xi_1 = \begin{bmatrix} A_i^T W_{1i} + C_i^T B_{Fi}^T & A_i^T W_{2i} + C_i^T B_{Fi}^T \\ A_{Fi}^T & A_{Fi}^T \end{bmatrix},$$
$$\Xi_2 = \begin{bmatrix} B_i^T W_{1i} + D_i^T B_{Fi}^T & B_i^T W_{2i} + D_i^T B_{Fi}^T \end{bmatrix},$$
$$\Xi_3 = \begin{bmatrix} L_i^T \\ -C_{Fi}^T \end{bmatrix}, \tilde{P}_i = \sum_{j=1}^{m} q_{ij} P_j - W_i - W_i^T,$$
$$P_i = \begin{bmatrix} P_{1i} & P_{2i} \\ * & P_{3i} \end{bmatrix}, W_i = \begin{bmatrix} W_{1i} & W_{2i} \\ W_{3i} & W_{3i} \end{bmatrix}.$$

*Proof* By Lemma 2.3, (2.32) holds if and only if there exist matrices $W_i$ such that

$$\begin{bmatrix} -P_i & 0 & \tilde{A}_i^T W_i & \tilde{C}_i^T \\ * & -\gamma^2 I & \tilde{B}_i^T W_i & 0 \\ * & * & \tilde{P}_i & 0 \\ * & * & * & -I \end{bmatrix} < 0. \tag{2.38}$$

Let $A_{Fi} = W_{3i}^T A_{fi}$, $B_{Fi} = W_{3i}^T B_{fi}$, $C_{Fi} = C_{fi}$, $P_i = \begin{bmatrix} P_{1i} & P_{2i} \\ * & P_{3i} \end{bmatrix}$ and $W_i = \begin{bmatrix} W_{1i} & P_{2i} \\ W_{3i} & W_{3i} \end{bmatrix}$. One sees that (2.38) is equivalent to (2.37). The proof is completed.

When the Markovian jump system approach is applied to the modeling and analysis of the NCSs, the transition probabilities should be known a prior. In some scenarios, the transition probabilities may be too expensive or even impossible to find. Then, the following switched linear system is more appropriate for such a situation.

## 2.4 Switched Systems

Since the switched system approach will be used to model and analyze the wireless networked systems in this book, some discussions on the switched systems are necessary. A switched system is a dynamical system that consists of a finite number of subsystems and a logical rule that orchestrates switching between these subsystems. These subsystems are usually described by a group of differential or difference equations. Unlike the Markovian jump systems in Sect. 2.2, the probabilities of the switching here are completely unknown. A simple way to classify switched systems is based on the dynamics of their subsystems, for example continuous-time or discrete-time, linear or nonlinear and so on. see Fig. 2.1.

In this book, we only consider the discrete-time switched linear systems as follows

$$x(k+1) = A_{\rho(k)} x(k), \tag{2.39}$$

where $\rho(k)$ is a switching signal and $\rho(k) \in \Omega = \{1, 2, \ldots, M\}$. The switching signal is usually piecewise constant and the subsystems are finite.

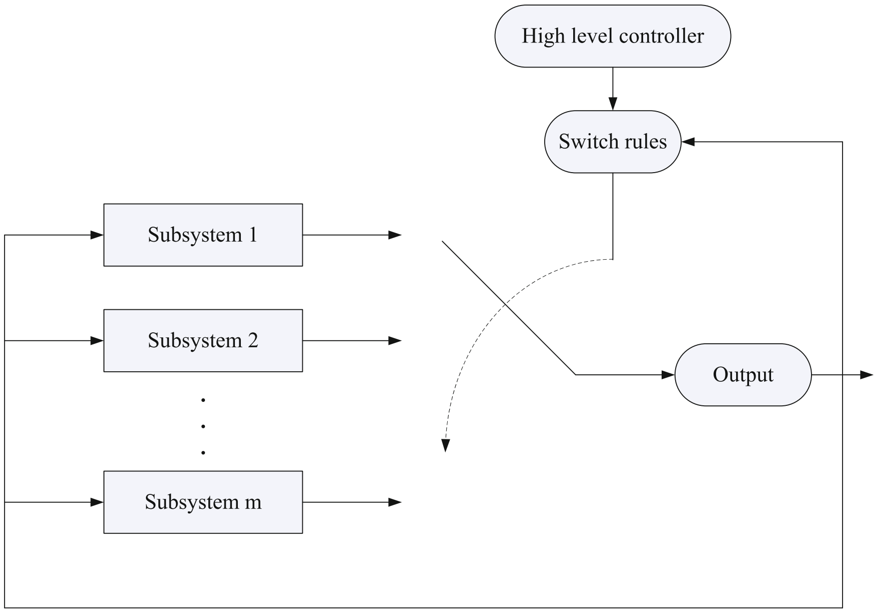


**Fig. 2.1** An illustrative example of switched system

The stability analysis of a switched system is more difficult than the traditional LTI system as the dynamics of a switched system is not only determined by its continuous-time or discrete-time dynamics but also by the switching signal. The interaction of continuous-time or discrete-time dynamics with the switching signal makes the research of switched system an attractive research direction. It is interesting to see that even all subsystems are stable, the overall switched system may be unstable [4]. In the last decades, many results have been reported in literature, and three basic problems are studied: (1) Stability analysis of switched systems under arbitrary switching; (2) Stability analysis of switched systems under some useful switching signal; (3) Constructing some switching signals such that the switched systems are stable. Among them, research of Problem 2 have been recently used to analyze the networked systems when they are modeled as a switched system. For problem 2, special attention has been paid on the stability analysis of switched systems under some special switching signals. For example, Morse [5] proposed a switching signal called as the dwell time switching signal for the switched system. They showed that the system is stable provided that the system dwell on each subsystem for a fixed time interval, called as the dwell time, i.e., the switching should be slow enough. Recently, Hespanha [6] relaxed the dwell time to the average dwell time such that only the average dwell time is required to be satisfied. More specifically, for any $k > k_0$, and a given switching signal $\rho(\tau)$, $k_0 \leq \tau \leq k$, let $N_\rho$ denote the number of switching of $\rho(\tau)$ over time interval $(k_0, k)$. If $N_\rho \leq N_0 + (k - k_0)\big/T_a$ holds for $T_a > 0$ and $N_0 \geq 0$, then $T_a$ is called the average dwell time and $N_0$ is the chatter bound, which is usually set to be zero. Hespanha [6] showed that the switched system is exponentially stable if the average dwell time of the successive switching is larger than a constant value. In this book, the average dwell time approach will be used in some chapters. Now some fundamental analysis and synthesis results are presented for discrete-time switched systems.

**Proposition 2.9** *For given scalars $0 < \lambda < 1$ and $\mu \geq 1$, the discrete-time switched linear system (2.39) is exponentially stable under the switching signal $\rho(k)$, if there exist a set of positive definite matrices $P_i$, such that the following inequalities*

$$\begin{bmatrix} -\lambda P_i & A_i^T P_i \\ * & -P_i \end{bmatrix} < 0, \tag{2.40}$$

$$P_i \leq \mu P_j, i \neq j, \tag{2.41}$$

$$T_a > T_a^* = -\frac{\ln \mu}{\ln \lambda}. \tag{2.42}$$

*are true for all $i, j \in \Omega$.*

*Proof* The switched system with average dwell time switching is known as a slowly switching system. We can define the switching time instant as $k_0 < k_1 < \cdots k_l < \cdots$. Then we construct the following piecewise Lyapunov function:

$$V_{\rho(k)}(x(k)) = x^T(k) P_{\rho(k)} x(k). \tag{2.43}$$

Then, it follows that for each $i = \rho(k)$,

$$\begin{aligned} & V_i(k+1) - \lambda V_i(k) \\ & \quad = x^T(k+1) P_i x(k+1) - \lambda x^T(k) P_i x(k) \\ & \quad = [A_i x(k)]^T P_i [A_i x(k)] - \lambda x^T(k) P_i x(k) \\ & \quad = x^T(k) \left[A_i^T P_i A_i - \lambda P_i\right] x(k). \end{aligned} \tag{2.44}$$

By (2.40), we have $A_i^T P_i A_i - \lambda P_i < 0$, which implies that $V_i(k+1) - \lambda V_i(k) < 0$. It is easy to see that

$$V_{\rho(k)}(k) \leq \lambda^{k-k_l} V_{\rho(k_l)}(k_l). \tag{2.45}$$

According to (2.41) and (2.45), and the switching sequence, we have

$$\begin{aligned} V_{\rho(k)}(k) &\leq \lambda^{k-k_l} V_{\rho(k_l)}(k_l) \\ &\leq \lambda^{k-k_l} \mu V_{\rho(k_{l-1})}(k_l) \\ &= \lambda^{k-k_l} \mu V_{\rho(k_l-1)}(k_l) \\ &\leq \cdots \\ &\leq \lambda^{k-k_0} \mu^{(k-k_0)/T_a} V_{\rho(k_0)}(k_0) \\ &= \left(\lambda \mu^{1/T_a}\right)^{(k-k_0)} V_{\rho(k_0)}(k_0). \end{aligned} \tag{2.46}$$

In addition, it follows from (2.43) that $V_{\rho(k)}(k) \geq \beta_1 \|x(k)\|^2$ and

$$V_{\rho(k)}(k) \leq \left(\lambda \mu^{1/T_a}\right)^{(k-k_0)} V_{\rho(k_0)}(k_0) \leq \left(\lambda \mu^{1/T_a}\right)^{(k-k_0)} \beta_2 \|x(k_0)\|^2, \tag{2.47}$$

which yields $\|x(k)\| \leq \sqrt{\beta_2/\beta_1} \beta^{(k-k_0)} \|x(k_0)\|^2$, where $\beta_1 = \min_{i \in \Omega} \lambda_{\min}(P_i)$, $\beta_2 = \max_{i \in \Omega} \lambda_{\max}(P_i)$ and $\beta = \lambda \mu^{1/T_a}$. Condition (2.42) guarantees $0 < \beta < 1$. Thus, the system (2.39) is exponentially stable.

*Remark 2.1* we have presented an exponential stability condition for system (2.39) in Proposition 2.9. Actually, there are also some other exponential stability conditions reported in recent literature, see [7]. In this book, however, Proposition 2.9 will play an important role in the sequential analysis.

For the controller design problem, we consider the following system:

$$\begin{cases} x(k+1) = A_{\rho(k)} x(k) + B_{\rho(k)} u(k), \\ u(k) = K_{\rho(k)} x(k), \end{cases} \tag{2.48}$$

where $u(k)$ is the control input and $K_{\rho(k)}$ is the feedback gain, which is to be determined. The following algorithm can be used to determine the controller gain.

**Proposition 2.10** *The discrete-time switched system (2.48) is exponentially stable if there exist positive definite matrices $Q_i$ and matrices $\bar{K}_i$ with appropriate dimensions, such that the (2.42) and following inequalities,*

$$\begin{bmatrix} -\lambda Q_i & Q_i A_i^T + \bar{K}_i^T B_i^T \\ * & -Q_i \end{bmatrix} < 0, \tag{2.49}$$

$$Q_j \leq \mu Q_i, i \neq j, \tag{2.50}$$

*hold for all $i \in \Omega$, moreover, the controller gain can be determined by $K_i = \bar{K}_i Q_i^{-1}$.*

*Proof* By replacing the $A_i$ by $A_i + B_i K_i$, (2.40) is written as

$$\begin{bmatrix} -\lambda P_i & (A_i + B_i K_i)^T P_i \\ * & -P_i \end{bmatrix} < 0. \tag{2.51}$$

By left- and right- multiplying diag$\{P_i^{-1}, P_i^{-1}\}$ and its transpose to (2.51), respectively, one sees that (2.51) is (2.49) by assigning $P_i^{-1} = Q_i$ and $\bar{K}_i^T = Q_i K_i^T$. By using similar manipulation, one can also have (2.50). The proof is completed.

Based on the filter design results in Sect. 2.2, one can also have the corresponding results for the switched systems, and the details are omitted. The fundamentals presented in this chapter will play an important role in the sequential chapters. Readers are encouraged to read this chapter and get to know how to analyze a Markovian jump system and a switched linear system.

## 2.5 Linear Matrix Inequalities

In the last decades, the linear matrix inequalities (LMIs) have emerged as a powerful tool to solve control and estimation problems that appear hard or even impossible to solve in an analytic way. Currently, several commercial or non-commercial software packages are available, with which an LMI problem can be easily solved by a simple coding. Since the main results of this book are given in terms of LMIs, some basic information is presented in this section. For more details on the LMI, we refer the readers to [1].

A typical LMI has the following form:

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0, \tag{2.52}$$

where $x_i \in \mathbb{R}^1$ is the scalar, and the symmetric matrices $F_i = F_i^T \in \mathbb{R}^{q\times q}$ are given. We call (2.52) is a strict LMI, while a non-strict LMI is $F(x) \geq 0$. In this section, we consider the strict LMI only. In the control system design, we will often encounter problems in which the variables are matrices, rather than the scalar in (2.52). We recall the results in Proposition 2.1, i.e., a continuous-time LTI system is asymptotic stable if the following inequality is true: $A^T P + PA < 0$, where $A \in \mathbb{R}^{n\times n}$ is the system matrix, and $P = P^T > 0$ is the unknown variable. The problem now is how to find a required matrix $P$ such that the inequality is true. Here, an illustrative example is first given to show how an LMI can be described by using the Matlab LMI toolbox.

For the above system, we can use "lmivar" and "lmiterm" to describe an LMI as follows:

```
setlmis([ ])
P=lmivar(1,[4 1]);
lmiterm([1 1 1 P],A',1);
lmiterm([1 1 1 P],1,A);
lmiterm([-2 1 1 P],1,1);
lmisys=getlmis
```

*Discussions*: An LMI usually starts by "setlmis" and ends by "getlmis". "setlmis" is a function, which is used to start a description of LMI. The function "lmivar" is used to define the unknown matrix variable $P$, and the function "lmiterm" is used to describe the detail of each LMI. "getlmis" returns to the internal LMI description of "lmisys", where it is also a name stored inside the computer.

Special attention should be paid on the "lmivar". A common description of this function is

$$P = \text{lmivar}(\text{type}, \text{struct}),$$

where "type" confirms the type of the matrix $P$, "struct" gives the structure of this variable. Usually it has three types:

- symmetric or diagonal.
- rectangular. Then, struct=($m$,$n$) describes the dimension.
- others.

*Example 2.1* Consider an LMI with three unknown matrices $P_1$, $P_2$ and $P_3$, where

- $P_1$ is a symmetric matrix, with dimension $4 \times 4$.
- $P_2$ is a rectangular matrix, with dimension $3 \times 4$.
- $P_3 = \begin{bmatrix} \Delta & 0 & 0 \\ 0 & \chi_1 & 0 \\ 0 & 0 & \chi_2 I_3 \end{bmatrix}$, where $\Delta$ is a symmetric matrix with dimension $3 \times 3$, $\chi_1$ and $\chi_2$ are two scalars, $I_3$ is an identity matrix with dimension $3 \times 3$.

The above matrices can be defined by the using "lmivar" as follows:

```
setlmis([ ])
P1=lmivar(1,[4 1]);
P2=lmivar(2,[3 4]);
P3=lmivar(1,[3 1;1 0;3 0]);
```

It should be pointed out that one needs to describe the LMI with the upper-triangular part only. For example, the following LMI,

$$\begin{bmatrix} A^T X + XA & XB \\ B^T X & -I \end{bmatrix} < 0, \tag{2.53}$$

can be described by

```
lmiterm([1 1 1 X],A',1);
lmiterm([1 1 1 X],1,A);
lmiterm([1 1 2 X],1,B);
lmiterm([1 2 2 0],-1);
```

Next we discuss how to determine whether such an LMI admit a solution. The LMI toolbox provides some solvers for LMI related problems such as

- Feasibility problem.
- Minimization problem subject to LMI constraint.
- Minimization problem of generalized eigenvalue.

In this book, we deal with the first two problems. Readers are referred to [1] for the third case. We now discuss how to solve the first two problems by the LMI toolbox.

The "feasp" solver is usually described by

$$[\text{tmin, xfeas}] = \text{feasp}\,(\text{lmisys, options, target})\,.$$

The feasibility problem is solvable, i.e., "lmisys" is feasible, provided that tmin $< 0$. When it is feasible, the "xfeas" gives a feasible solution to the decision variable. "target" is introduced for the target value of "tmin" such that "tmin $<$ target", then the searching process ends. Usually, target $= 0$ is used.

Consider an LTI system $\dot{x}(t) = Ax(t) + Bu(t)$ with the state matrices as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 11 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

We first check the stability of this system when $u = 0$. By running the following code:

```
A=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0];
setlmis([ ])
P=lmivar(1,[4 1]);
lmiterm([1 1 1 P],A',1);
lmiterm([1 1 1 P],1,A);
lmiterm([-2 1 1 P],1,1);
lmisys=getlmis;
[tmin, xfeas] = feasp(lmisys)
PP=dec2mat(lmisys,xfeas,P)
```

we have

$$\text{tmin} = 1.1659e - 15,$$

which means that it is not stable. Now we use Proposition 2.3 to determine the feedback controller gain $K$. Based on the following code:

```
A=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0];
B=[0;1;0;1];
X=lmivar(1,[4 1]);
Y=lmivar(2,[1 4]);
lmiterm([1 1 1 X],A,1);
lmiterm([1 1 1 X],1,A');
lmiterm([1 1 1 Y],B,-1);
lmiterm([1 1 1 -Y],-1,B');
lmiterm([-2 1 1 X],1,1);
lmisys=getlmis;
[tmin, xfeas] = feasp(lmisys)
XX=dec2mat(lmisys,xfeas,X);
YY=dec2mat(lmisys,xfeas,Y);
K=YY*inv(XX)
```

We obtain

$$K = \begin{bmatrix} -3.0056 & -5.8780 & 61.8612 & 13.0596 \end{bmatrix}.$$

In control system design, we may encounter some optimization problem, e.g., the disturbance attenuation level should be minimized in the $H_\infty$ control and estimation problem. The "mincx" solver is usually adopted to the optimization problem. For example, the following optimization problem:

$$\begin{aligned} &\min c^T x \\ &s.t.\ A(x) < B(x). \end{aligned} \tag{2.54}$$

The "mincx" solver is described by

$$[\text{copt}, \text{xopt}] = \text{mincx}\ (\text{lmisys}, c, \text{options}, \text{xinit}, \text{target})\,.$$

As in the "feasp" solver, "mincx" returns the optimal value to "xopt", which can be outputted by using the "dec2mat" function. Traditionally, "lmisys" and "$c$" are compulsory for the "mincx" solver, while the rest is not. We now take a numerical example to illustrate how to use the "mincx" solver.

*Example 2.2* Consider the following optimization problem:

$$\begin{aligned} &\min \mathrm{Tr(P)} \\ &s.t.\ A^T P + PA + PBB^T P + Q < 0, \end{aligned} \tag{2.55}$$

where $P = P^T$ is a unknown matrix and

$$A = \begin{bmatrix} -1 & -2 & 1 \\ 1 & 2 & 1 \\ 1 & -2 & -1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$
$$Q = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -3 & -12 \\ 0 & -12 & -36 \end{bmatrix}.$$

*Solution*: With Lemma 2.1, we formulate the following optimization problem:

$$\begin{aligned} & \min \mathrm{Tr(P)} \\ s.t. & \begin{bmatrix} A^T P + PA + Q & PB \\ B^T P & -I \end{bmatrix} < 0. \end{aligned} \tag{2.56}$$

A possible coding is given by:

```
A=[-1 -2 1;1 2 1;1 -2 -1];
B=[1;0;1];
Q=[1 -1 0;-1 -3 -12;0 -12 -36];
setlmis([ ])
P=lmivar(1,[3 1]);
lmiterm([1 1 1 P],A',1);
lmiterm([1 1 1 P],1,A);
lmiterm([1 1 1 0],Q);
lmiterm([1 1 2 P],1,B);
lmiterm([1 2 2 0],-1);
LMIs=getlmis;
c=mat2dec(LMIs,eye(3));
options=[le-5,0,0,0,0];
[copt, xopt]=mincx(LMIs,c,options);
xopt1=dec2mat(LMIs,xopt,P)
```

It gives the obtained optimal $P$ as

$$\begin{aligned} & \mathrm{xopt1} = \\ & \begin{matrix} -3.9278 & -12.3556 & -0.0722 \\ -12.3556 & -39.5667 & 1.5000 \\ -0.07221 & 1.5000 & -4.7834 \end{matrix} \end{aligned}$$

## References

1. S. Boyd, L. Ghaoui, E. Feron, V. Balakrishnan, *Linear Matri Inequalities in Systems and Control Theory* (SIAM, Philadelphia, 1994)

2. L.H. Xie, C.E. de. Souza, Y.Y. Wang, Robust control of discrete-time uncertain dynamical systems. Automatica. **29**(4), 1133–1137 (1993)
3. M.C. de Oliveira, J. Bernussou, J.C. Geromel, A new discrete-time robust stability condition. Syst. Control lett. **37**(4), 261–265 (1999)
4. D. Liberzon, *Switching in Systems and Control* (Birkhauser, Boston, 2003)
5. A.S. Morse, Supervisory control of families of linear set-point controllers-part 1: exact matching. IEEE Trans. Autom. Control **41**(10), 1413–1431 (1996)
6. J. Hespanha, A. Morse, Stability of switched systems with average dwell time, in *Proceedings of the 38th IEEE Conference on Decision and Control* (1999), pp. 2655–2660
7. H. Lin, P.J. Antsaklis, Stability and stabilizability of switched linear systems: a survey of recent results. IEEE Trans. Autom. Control **54**(2), 308–322 (2009)

http://www.springer.com/978-3-319-53122-9

Filtering and Control of Wireless Networked Systems
Zhang, D.; Wang, Q.-G.; Yu, L.
2017, XI, 232 p. 80 illus., 50 illus. in color., Hardcover
ISBN: 978-3-319-53122-9