Chapter 4
Optimal Integrated Control and Scheduling of Resource-Constrained Systems

The general model of resource-constrained systems allows the on-line assignment of the sensors-to-controller and the controller-to-actuators scheduling vectors. This assignment may be based on a pre-computed off-line schedule or on an on-line scheduling algorithm. The problem of the optimal integrated control and *off-line* scheduling of the sensors-to-controller link is the dual problem of the optimal integrated control and off-line scheduling of the controller-to-actuators link, and may be solved in a similar way. However, the problem of the optimal integrated control and *on-line* scheduling of the sensors-to-controller link is different from the problem of the optimal integrated control and on-line scheduling of the controller-to-actuators link, and its formulation and solving are extremely dependant on the used technology (possibility of on-line arbitration based on the comparison of messages identifiers like in *CAN* networks for example). Furthermore, it is not clear whether an on-line scheduling algorithm of the sensors-to-controller link will be better (from a control performance point of view) than a predefined off-line scheduling algorithm, especially, because the sensors are physically distributed whereas the optimal on-line assignment of the sensors-to-controller link requires the knowledge of all sensors values, which may not be possible due to the communication constraints. To the best of the authors knowledge, this last question remains an open problem.

For that reason, in this chapter, as well as in the forthcoming Chaps. 5 and 6, we will assume that the state of the plant is available to the controller at each sampling period. This assumption will allow us to mainly focus on the problem of the optimal integrated control and scheduling of the controller-to-actuators link. Note that assuming that the state of the plant is available to the controller at each sampling period does not necessarily mean that we are restricted to a particular architecture, but rather that "state of the art methods", such as an off-line scheduling, are deployed to obtain a satisfactory estimate of the state at the controller, using an adequate part of the bandwidth. Let $\mathscr{S}$ be a resource-constrained system verifying this assumption. Furthermore, $\mathscr{S}$ verifies $p = n = b_r$, $\sigma(k) = 1_{n,1}$, $\forall k \in \mathbb{N}$ and $0 < b_w \leq m$. To simplify the notation, let $b = b_w$. In the model considered in the sequel, the resource-constrained system $\mathscr{S}$ has two types of inputs: control inputs and scheduling inputs

A. Çela et al., *Optimal Design of Distributed Control and Embedded Systems*, Communications and Control Engineering, DOI 10.1007/978-3-319-02729-6_4,

of the controller-to-actuators link. The fundamental problem that will be addressed is the problem of the joint optimization of control and scheduling. This problem may be viewed as the generalization of optimal control problems for linear sampled-data system. It naturally appears as having a hybrid character, since continuous aspects (system dynamics) as well as logical aspects (scheduling) interact.

Using the basic results of optimal control theory, we first describe the solution of the optimal control problem given a fixed communication sequence, over finite and infinite horizons. Then the problem of the optimal integrated control and scheduling of resource-constrained systems is formulated and solved. The formulation and solving of this problem are based on the existing theoretical tools from hybrid systems theory and especially the *MLD* framework [21], where this problem may be perfectly modeled. Finally, based on a numerical example, the solutions of this problem are studied.

4.1 Performance Index Definition

In order to introduce an appropriate measure of the *"quality"* of the control and scheduling, inspired by optimal control metrics [14], a quadratic cost function is associated to the system (3.2) (and implicitly to the resource-constrained system $\mathscr{S}$).

$$J_c(x_c, u_c, 0, T_f) = \int_0^{T_f} \left(x_c^T(t) Q_c x_c(t) + u_c^T(t) R_c u_c(t)\right) dt + x_c^T(T_f) S_c x_c(T_f) \tag{4.1}$$

where $T_f = NT_s$ and Q_c, R_c and S_c are positive definite matrices. These matrices define the design specifications of the ideal continuous-time controller. The sampled-data representation of the cost function $J_c(x_c, u_c, 0, T_f)$ at the sampling period T_s is

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N) Q_0 x(N). \tag{4.2}$$

The expressions of Q_1, Q_2, Q_{12} and Q_0 may be found in ([14], pp. 411–412). In the following, it is assumed that Q, Q_2 and Q_0 are positive definite matrices, where

$$Q = \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}.$$

Note that this representation does not involve any approximation and it is exact. Using this representation, the inter-sample behavior is taken into account.

4.2 Optimal Control over a Finite Horizon for a Fixed Communication Sequence

The problem of the optimal control, over a finite-time N, for a fixed admissible and maximal communication sequence δ^{N-1}, may be formulated as follows:

Problem 4.1 Given an initial state $x(0)$ and a final time N, find the optimal control sequence $v^{N-1} = (v(0), \dots, v(N-1))$ that minimizes the cost function:

$$J(x,u,0,N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N) Q_0 x(N),$$

subject to

$$\begin{aligned} & x(k+1) = Ax(k) + Bu(k), \\ & u_i(k) = v_j(k), \quad \text{if } \delta_i(k) = 1 \text{ and } \sum_{l=1}^{i} \delta_l(k) = j, \\ & u_i(k) = u_i(k-1), \quad \text{otherwise.} \end{aligned}$$

In order to solve this problem, we reconsider the state representation (3.12) of system $\mathscr{S}$, which was established in Sect. 3.4 of Chap. 3. For a given maximal controller-to-actuators scheduling δ, system $\mathscr{S}$ is described by the state equation (3.12a). The cost function $J(x,u,0,N)$ may be rewritten in the form

$$J(x,u,0,N) = J(\tilde{x},v,0,N) = \sum_{k=0}^{N-1} \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix} + \tilde{x}^T(N) \tilde{Q}_0 \tilde{x}(N),$$

where

$$\tilde{Q}(k) = \begin{bmatrix} Q_1 & Q_{12} E_\delta(k) & Q_{12} D_\delta(k) \\ E_\delta^T(k) Q_{12}^T & E_\delta^T(k) Q_2 E_\delta(k) & E_\delta^T(k) Q_2 D_\delta(k) \\ D_\delta^T(k) Q_{12}^T & D_\delta^T(k) Q_2 E_\delta(k) & D_\delta^T(k) Q_2 D_\delta(k) \end{bmatrix},$$

and

$$\tilde{Q}_0 = \begin{bmatrix} Q_0 & 0_{n,m} \\ 0_{m,n} & 0_{m,m} \end{bmatrix}.$$

Problem 4.1 is equivalent to the optimal control problem of a discrete linear time-varying system, given as follows:

Problem 4.2 Given an initial state $x(0)$ and a final time N, find the optimal control sequence $v^{N-1} = (v(0), \ldots, v(N-1))$ that minimizes the cost function

$$J(\tilde{x}, v, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix} + \tilde{x}^T(N)\tilde{Q}_0\tilde{x}(N),$$

subject to

$$\tilde{x}(k+1) = \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)v(k). \tag{4.3}$$

Problem 4.2 is a classical optimal control problem of a discrete linear time-varying system. Different methods for its resolution were developed in control textbooks (see, for instance, [14]). Let:

$$\tilde{Q}_1(k) = \begin{bmatrix} Q_1 & Q_{12}E_\delta(k) \\ E_\delta^T(k)Q_{12}^T & E_\delta^T(k)Q_2E_\delta(k) \end{bmatrix},$$

$$\tilde{Q}_{12}(k) = \begin{bmatrix} Q_{12}D_\delta(k) \\ E_\delta^T(k)Q_2D_\delta(k) \end{bmatrix},$$

and

$$\tilde{Q}_2(k) = D_\delta^T(k)Q_2D_\delta(k).$$

The solution of this problem closely depends on the solution of an algebraic equation, involving the variable $\tilde{S}(k)$ and described by

$$\begin{aligned}\tilde{S}(k) = {} & \tilde{A}^T(k)\tilde{S}(k+1)\tilde{A}(k) + \tilde{Q}_1(k) - \big(\tilde{A}^T(k)\tilde{S}(k+1)\tilde{B}(k) + \tilde{Q}_{12}(k)\big) \\ & \times \big(\tilde{B}^T(k)\tilde{S}(k+1)\tilde{B}(k) + \tilde{Q}_2(k)\big)^{-1}\big(\tilde{B}^T(k)\tilde{S}(k+1)\tilde{A}(k) + \tilde{Q}_{12}^T(k)\big)\end{aligned} \tag{4.4}$$

under the terminal condition

$$\tilde{S}(N) = \tilde{Q}_0. \tag{4.5}$$

Equation (4.4) is the discrete algebraic Riccati equation associated to the Problem 4.2. Knowing that $\tilde{Q}_0$ is semi-definite positive and $\tilde{Q}_2(k)$ is definite positive (because $D_\delta(k)$ is injective when δ is a maximal admissible communication sequence), than this equation admits a unique positive semi-definite solution. Consequently, Problem 4.2 admits a unique solution [14] defined by

$$v(k) = -\tilde{K}(k)\tilde{x}(k), \tag{4.6}$$

with

$$\tilde{K}(k) = \big(\tilde{Q}_2(k) + \tilde{B}^T(k)\tilde{S}(k+1)\tilde{B}(k)\big)^{-1}\big(\tilde{B}^T(k)\tilde{S}(k+1)\tilde{A}(k) + \tilde{Q}_{12}^T(k)\big). \tag{4.7}$$

4.3 Optimal Control over an Infinite Horizon for a Fixed Communication Sequence

Consider a T-periodic maximal communication sequence δ^{T-1} defined by

$$\delta^{T-1} = \big(\delta(0), \dots, \delta(T-1)\big)$$

and verifying $\delta(k+T) = \delta(k)$. Assume furthermore that $\delta \in \mathscr{S}^c$, where $\mathscr{S}^c$ is the set of communication sequences that guarantee the reachability of system (3.12). The periodicity of the communication sequence induces the periodicity of the resource-constrained system $\mathscr{S}$. As a result, matrices $\tilde{A}(k)$, $\tilde{B}(k)$ and $\tilde{Q}(k)$ satisfy $\tilde{A}(k+T) = \tilde{A}(k)$, $\tilde{B}(k+T) = \tilde{B}(k)$ and $\tilde{Q}(k) = \tilde{Q}(k+T)$.

Let ι be a discrete time instant and H a positive integer, assume that $N = \iota + HT$ and consider the optimal control problem:

$$\begin{cases} \min\limits_{v} \quad J(\tilde{x}, v, \iota, N) = \displaystyle\sum_{k=\iota}^{N-1} \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix} + \tilde{x}^T(N)\tilde{Q}_0\tilde{x}(N) \\ \text{subject to} \quad \tilde{x}(k+1) = \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)v(k). \end{cases} \tag{4.8}$$

As illustrated in [35], a time invariant reformulation of the optimal control problem (4.8) may be obtained by using the lifting technique. The time invariant reformulation may be seen as a down sampled representation of system (3.12) with periodicity T, having an augmented input vector. In the following, the formulation of the time invariant representation is described.

Let Φ be the transition matrix associated with the state matrix $\tilde{A}$ defined by:

$$\begin{cases} \Phi(l,s) = \tilde{A}(l-1)\tilde{A}(l-2)\cdots\tilde{A}(s) & \text{if } l > s, \\ \Phi(l,l) = I_{n+m}. \end{cases}$$

Let Γ the matrix defined for $s < l < s+T$ by

$$\Gamma(l,s) = \Big[\Phi(l,s+1)\tilde{B}(s)\ \Phi(l,s+2)\tilde{B}(s+1)\ \cdots\ \Phi(l,l)\tilde{B}(l-1)\ \underbrace{0_{n+m,b}\ \cdots\ 0_{n+m,b}}_{T-l-s}\Big]$$

and for $s = l$ by

$$\Gamma(s,s) = [0_{n+m,b} \quad \cdots \quad 0_{n+m,b}].$$

Let

$$\bar{x}_\iota(q) = \tilde{x}(\iota + qT),$$

and

$$\bar{v}_\iota(q) = \begin{bmatrix} v(\iota + qT) \\ \vdots \\ v(\iota + (q+1)T - 1) \end{bmatrix},$$

then for $0 \leq i \leq T$:

$$\tilde{x}(\iota + qT + i) = \Phi(\iota + i, \iota)\bar{x}_\iota(q) + \Gamma(\iota + i, \iota)\bar{v}_\iota(q).$$

In particular, let

$$\bar{A}_\iota = \Phi(\iota + T, \iota),$$

and

$$\bar{B}_\iota = \Gamma(\iota + T, \iota),$$

then the following relation is obtained:

$$\bar{x}_\iota(q + 1) = \bar{A}_\iota \bar{x}_\iota(q) + \bar{B}_\iota \bar{v}_\iota(q).$$

Let $\Lambda(i)$ the matrix defined for $0 \leq i < T$ by

$$\Lambda(i) = \left[\underbrace{0_{b,b} \quad \cdots \quad 0_{b,b}}_{i} \, I_b \quad \underbrace{0_{b,b} \quad \cdots \quad 0_{b,b}}_{T-i-1} \right],$$

then the cost function may be written

$$J(\tilde{x}, v, \iota, N) = J(\bar{x}_\iota, \bar{v}_\iota, 0, H) = \sum_{q=0}^{H-1} \begin{bmatrix} \bar{x}_\iota(q) \\ \bar{v}_\iota(q) \end{bmatrix}^T \bar{Q}_\iota \begin{bmatrix} \bar{x}_\iota(q) \\ \bar{v}_\iota(q) \end{bmatrix} + \bar{x}_\iota^T(H)(\bar{Q}_\iota)_0 \bar{x}_\iota(H)$$

where

$$\bar{Q}_\iota = \sum_{i=0}^{T-1} F^T(i)\tilde{Q}(\iota + i)F(i),$$

$$F(i) = \begin{bmatrix} \Phi(\iota + i, \iota) & \Gamma(\iota + i, \iota) \\ 0_{b,n+m} & \Lambda(i) \end{bmatrix}$$

and $(\bar{Q}\iota)_0 = \tilde{Q}_0$. Finally, the following optimal control problem is obtained

$$\begin{cases} \min\limits_{\bar{v}_\iota} \quad J(\bar{x}_\iota, \bar{v}_\iota, 0, H) = \bar{x}_\iota^T(H)(\bar{Q}\iota)_0 \bar{x}_\iota(H) + \sum\limits_{q=0}^{H-1} \begin{bmatrix} \bar{x}_\iota(q) \\ \bar{v}_\iota(q) \end{bmatrix}^T \bar{Q}_\iota \begin{bmatrix} \bar{x}_\iota(q) \\ \bar{v}_\iota(q) \end{bmatrix} \\ \text{subject to} \quad \bar{x}_\iota(q + 1) = \bar{A}_\iota \bar{x}_\iota(q) + \bar{B}_\iota \bar{v}_\iota(q). \end{cases} \tag{4.9}$$

The corresponding discrete algebraic Riccati equation is given by:

$$\begin{aligned} \bar{S}_\iota(q) = {} & \bar{A}_\iota^T \bar{S}_\iota(q + 1)\bar{A}_\iota + \bar{Q}_{\iota_1} - \left(\bar{A}_\iota^T \bar{S}_\iota(q + 1)\bar{B}_\iota + \bar{Q}_{\iota_{12}}\right) \\ & \times \left(\bar{B}_\iota^T \bar{S}_\iota(q + 1)\bar{B}_\iota + \bar{Q}_{\iota_2}(q)\right)^{-1}\left(\bar{B}_\iota^T \bar{S}_\iota(q + 1)\bar{A}_\iota + \bar{Q}_{\iota_{12}}^T\right), \end{aligned} \tag{4.10}$$

with the terminal condition

$$\bar{S}_\iota(H) = (\bar{Q}_\iota)_0, \tag{4.11}$$

where

$$\bar{Q}_\iota = \begin{bmatrix} \bar{Q}_{\iota_1} & \bar{Q}_{\iota_{12}} \\ \bar{Q}_{\iota_{12}}^T & \bar{Q}_{\iota_2} \end{bmatrix}.$$

The relationship between the solutions of Riccati equations (4.4) and (4.10), which are respectively, associated to the optimal control problems (4.8) and (4.9), is formalized by the following result.

Lemma 4.1 [35] *If* $\bar{S}_\iota(H) = \tilde{S}(\iota + NT) = \tilde{Q}_0$, *then* $\bar{S}_\iota(q) = \tilde{S}(\iota + qT)$ *for all* $q \leq H$.

This result follows from the fact that optimal control problems (4.8) and (4.9) are similar. By imposing the terminal condition $\bar{S}_\iota(H) = \tilde{S}(\iota + NT) = \tilde{Q}_0$, called *periodic generator*, the optimal costs must be identical, which implies that the solutions of the Riccati equations (4.4) and (4.10) must be the same. As a result, when $H \to +\infty$, $\bar{S}_\iota(q)$ converges to a constant solution $\bar{S}_\iota$. Consequently, $\tilde{S}(k)$ converges to a periodic solution, defined by $\tilde{S}(k) = \bar{S}_{(k \bmod T)}$. This periodic solution may be obtained by solving the algebraic Riccati equation associated with problem (4.9) when $H \to +\infty$ and for $\iota \in \{1, \ldots, T\}$. The optimal control gains may then be deduced from the relation (4.7) and are described by the sequence $(\tilde{K}(0), \ldots, \tilde{K}(T-1))$.

This formulation will be of practical importance at the next chapters when the problems of the optimal integrated control and scheduling of resource-constrained systems will be tackled.

4.4 Finite-Time Optimal Integrated Control and Scheduling

In this paragraph, the problem of the finite-time optimal control and scheduling is formulated and translated into the mixed-integer quadratic programming (MIQP) formulation. We assume in the following that $u(k) = 0$ and $v(k) = 0$ for $k < 0$, and that control commands $u(k)$ and $v(k)$ are subject to saturation constraints

$$\begin{cases} L_i \leq u_i(k) \leq U_i, \\ L_i \leq v_i(k) \leq U_i \end{cases} \tag{4.12}$$

where $L_i < 0$ and $U_i > 0$.

4.4.1 Problem Formulation

The finite-time optimal control and scheduling problem may be formalized as follows:

Problem 4.3 Given an initial state $x(0)$ and a final time N, find the optimal control sequence $v^{N-1} = (v(0), \ldots, v(N-1))$ as well as the optimal communication sequence $\delta^{N-1} = (\delta(0), \ldots, \delta(N-1))$ which minimizes the performance index:

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N) Q_0 x(N)$$

subject to:

$$\begin{aligned}
&x(k+1) = Ax(k) + Bu(k), \\
&\sum_{i=1}^{m} \delta_i(k) = b, \\
&u_i(k) = v_j(k), \quad \text{if } \delta_i(k) = 1 \text{ and } \sum_{l=1}^{i} \delta_l(k) = j, \\
&u_i(k) = u_i(k-1), \quad \text{otherwise}, \\
&L_i \le v_i(k) \le U_i.
\end{aligned}$$

The problem of finding the optimal control sequence v^{N-1} for a given fixed communication sequence δ^{N-1} is a quadratic programming (*QP*) problem. The number of possible communication sequences is finite. The resolution of Problem 4.3 may be reduced to the exploration of all the feasible maximal communication sequences and the solving of a *QP* problem for each fixed communication sequence. However, in practice, the number of feasible communication sequences grows exponentially with N, which means that exhaustive search may not be applied to problems with large values of N.

The solution of the Problem 4.3 may be obtained by solving a simpler optimization problem, which may be seen as a constrained control problem, where the variables v^{N-1} are eliminated and the constraint (3.8) is replaced by (3.7). Let $u^{N-1} = (u(0), \ldots, u(N-1))$, this problem may be stated as follows:

Problem 4.4 Given an initial state $x(0)$ and a final time N, find the optimal control sequence u^{N-1} as well as the optimal scheduling sequence δ^{N-1} that minimizes the performance index:

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N) Q_0 x(N),$$

subject to

$$\begin{aligned}
&x(k+1) = Ax(k) + Bu(k),\\
&\sum_{i=1}^{m} \delta_i(k) = b,\\
&\delta_i(k) = 0 \quad \Longrightarrow \quad u_i(k) = u_i(k-1),\\
&L_i \le v_i(k) \le U_i.
\end{aligned}$$

We observe that the constraints of optimal scheduling problem are composed of a set of linear equalities and inequalities as well as of the following logical formula:

$$\delta_i(k) = 0 \quad \Longrightarrow \quad u_i(k) = u_i(k-1). \tag{4.13}$$

In order to solve this problem, it is necessary to translate the logical formula (4.13) into linear inequalities. The connective "$\Longrightarrow$" may be eliminated if (4.13) is rewritten in the following equivalent form

$$u_i(k) - u_i(k-1) = \delta_i(k)u_i(k) - \delta_i(k)u_i(k-1). \tag{4.14}$$

However, Eq. (4.14) contains terms which are the product of logical variables and continuous variables. Using the procedure described in [21], this product is translated into an equivalent conjunction of linear inequalities. For example, let

$$\xi_i(k) = \delta_i(k)u_i(k). \tag{4.15}$$

Then (4.14) may be rewritten in the equivalent form:

$$\begin{cases}
\xi_i(k) \le U_i\delta_i(k),\\
\xi_i(k) \ge L_i\delta_i(k),\\
\xi_i(k) \le u_i(k) - L_i\bigl(1-\delta_i(k)\bigr),\\
\xi_i(k) \ge u_i(k) - U_i\bigl(1-\delta_i(k)\bigr).
\end{cases} \tag{4.16}$$

Note that the same procedure may be applied to

$$o_i(k) = \delta_i(k)u_i(k-1). \tag{4.17}$$

Let

$$\check{\Delta} = \begin{bmatrix} \delta(0) \\ \vdots \\ \delta(N-1) \end{bmatrix}; \qquad \check{U} = \begin{bmatrix} u(0) \\ \vdots \\ u(N-1) \end{bmatrix},$$

$$\check{X} = \begin{bmatrix} x(0) \\ \vdots \\ x(N) \end{bmatrix}; \qquad \check{\varXi} = \begin{bmatrix} \xi(0) \\ \vdots \\ \xi(N-1) \end{bmatrix},$$

and

$$\check{O} = \begin{bmatrix} o(0) \\ \vdots \\ o(N-1) \end{bmatrix}; \qquad \mathscr{V} = \begin{bmatrix} \check{\varDelta} \\ \check{U} \\ \check{X} \\ \check{\varXi} \\ \check{O} \end{bmatrix},$$

then Problem 4.4 may be written in the form

$$\begin{cases} \min\limits_{\mathscr{V}} \quad \frac{1}{2}\mathscr{V}^T \mathscr{F} \mathscr{V} + \mathscr{G}^T \mathscr{V} \\ \mathscr{A}\mathscr{V} \le \mathscr{B}, \\ \mathscr{V}_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, mN\}, \end{cases} \tag{4.18}$$

where the expressions $\mathscr{F}$, $\mathscr{G}$, $\mathscr{A}$ and $\mathscr{B}$ are defined in the sequel.

Matrices $\mathscr{A}$ and $\mathscr{B}$ include both the equality and the inequality constraints of the problem, and may be written in the form:

$$\mathscr{A} = \begin{bmatrix} \mathscr{A}_{\text{eq}} \\ -\mathscr{A}_{\text{eq}} \\ \mathscr{A}_{\text{in}} \end{bmatrix},$$

$$\mathscr{B} = \begin{bmatrix} \mathscr{B}_{\text{eq}} \\ -\mathscr{B}_{\text{eq}} \\ \mathscr{B}_{\text{in}} \end{bmatrix}.$$

It easy to see that the relation

$$\mathscr{A}_{\text{eq}}\mathscr{V} = \mathscr{B}_{\text{eq}}$$

is equivalent to

$$(\mathscr{A}_{\text{eq}}\mathscr{V} \le \mathscr{B}_{\text{eq}}) \wedge (-\mathscr{A}_{\text{eq}}\mathscr{V} \le -\mathscr{B}_{\text{eq}}).$$

Matrices $\mathscr{A}_{\text{eq}}$ and $\mathscr{B}_{\text{eq}}$ describe equalities (3.4a), (4.14) and impose to the scheduling sequence of the controller-to-actuators link to be maximal. Matrix $\mathscr{A}_{\text{eq}}$ is defined by

$$\mathscr{A}_{\text{eq}} = \begin{bmatrix} SC & 0_{N,n(N+1)} \; 0_{N,mN} & 0_{N,mN} & 0_{N,mN} \\ 0_{n(N+1),mN} & ST & 0_{n(N+1),mN} & 0_{n(N+1),mN} \\ 0_{mN,mN} & 0_{mN,n(N+1)} \; UU & -I_{mN} & I_{mN} \end{bmatrix}$$

where

- SC is a $N \times mN$ matrix described by

$$SC = \begin{bmatrix} 1_{1,m} & 0_{1,m} & \cdots & 0_{1,m} \\ 0_{1,m} & 1_{1,m} & \cdots & 0_{1,m} \\ & & \ddots & \\ 0_{1,m} & 0_{1,m} & \cdots & 1_{1,m} \end{bmatrix}.$$

- ST is a $n(N+1) \times (n(N+1)+mN)$ matrix that is used to represent the constraint (3.4a):

$$ST = \begin{bmatrix} I_n & 0_{n,n} & 0_{n,n} & \cdots & 0_{n,n} & 0_{n,m} & 0_{n,m} & \cdots & 0_{n,m} \\ -A & I_n & 0_{n,n} & \cdots & 0_{n,n} & -B & 0_{n,m} & \cdots & 0_{n,m} \\ 0_{n,n} & -A & I_n & \cdots & 0_{n,n} & 0_{n,m} & -B & \cdots & 0_{n,m} \\ & & \ddots & \ddots & & & & \ddots & \\ 0_{n,n} & 0_{n,n} & & -A & I_n & 0_{n,m} & 0_{n,m} & \cdots & -B \end{bmatrix}.$$

- UU is a $mN \times mN$ matrix described by

$$UU = \begin{bmatrix} I_m & 0_{m,m} & 0_{m,m} & \cdots & 0_{m,m} \\ -I_m & I_m & 0_{m,m} & \cdots & 0_{m,m} \\ 0_{m,m} & -I_m & I_m & \cdots & 0_{m,m} \\ & & \ddots & \ddots & \\ 0_{m,m} & 0_{m,m} & & -I_m & I_m \end{bmatrix}.$$

Matrix $\mathscr{B}_{\text{eq}}$ is defined by

$$\mathscr{B}_{\text{eq}} = \begin{bmatrix} b \times 1_{N,1} \\ x(0) \\ 0_{nN,1} \\ 0_{mN,1} \end{bmatrix}.$$

Let U and L the vectors defined by

$$U = \begin{bmatrix} U_1 \\ \vdots \\ U_m \end{bmatrix}; \qquad L = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix}.$$

Matrices $\mathscr{A}_{\text{in}}$ and $\mathscr{B}_{\text{in}}$ describe the constraints defining the variables $\xi(k)$ and $o(k)$ ((4.15) and (4.17)). Matrix $\mathscr{A}_{\text{in}}$ is defined by

$$\mathscr{A}_{\text{in}} = \begin{bmatrix} -\operatorname{Diag}(U) & 0_{mN,n(N+1)} & 0_{mN,mN} & I_{mN} & 0_{mN,mN} \\ \operatorname{Diag}(L) & 0_{mN,n(N+1)} & 0_{mN,mN} & -I_{mN} & 0_{mN,mN} \\ -\operatorname{Diag}(L) & 0_{mN,n(N+1)} & -I_{mN} & I_{mN} & 0_{mN,mN} \\ \operatorname{Diag}(U) & 0_{mN,n(N+1)} & I_{mN} & -I_{mN} & 0_{mN,mN} \\ -\operatorname{Diag}(U) & 0_{mN,n(N+1)} & 0_{mN,mN} & 0_{mN,mN} & I_{mN} \\ \operatorname{Diag}(L) & 0_{mN,n(N+1)} & 0_{mN,mN} & 0_{mN,mN} & -I_{mN} \\ -\operatorname{Diag}(L) & 0_{mN,n(N+1)} & -UM & 0_{mN,mN} & I_{mN} \\ \operatorname{Diag}(U) & 0_{mN,n(N+1)} & UM & 0_{mN,mN} & -I_{mN} \end{bmatrix}$$

where

$$UM = -(UU - I_{mN}).$$

Next, the matrix $\mathscr{B}_{\text{in}}$ is defined by

$$\mathscr{B}_{\text{in}} = \begin{bmatrix} 0_{m,1} \\ 0_{m,1} \\ -L \\ U \\ 0_{m,1} \\ 0_{m,1} \\ -L \\ U \end{bmatrix}.$$

Finally, matrices $\mathscr{F}$ and $\mathscr{G}$ are defined by

$$\mathscr{F} = \begin{bmatrix} 0_{mN,mN} & 0_{mN,n(N+1)+mN} & 0_{mN,2mN} \\ 0_{n(N+1)+mN,mN} & \check{\mathscr{Q}} & 0_{n(N+1)+mN,2mN} \\ 0_{2mN,mN} & 0_{2mN,n(N+1)+mN} & 0_{2mN,2mN} \end{bmatrix}$$

and respectively,

$$\mathscr{G} = 0_{(4m+(n+1))N,1}$$

where

$$\check{\mathscr{Q}} = \begin{bmatrix} Q_1 & 0_{n,n} & \cdots & 0_{n,n} & 0_{n,n} & Q_{12} & 0_{n,m} & \cdots & 0_{n,m} \\ 0_{n,n} & Q_1 & \cdots & 0_{n,n} & 0_{n,n} & 0_{n,m} & Q_{12} & \cdots & 0_{n,m} \\ & & \ddots & & & & & \ddots & \\ 0_{n,n} & 0_{n,n} & \cdots & Q_1 & 0_{n,n} & 0_{n,m} & 0_{n,m} & \cdots & Q_{12} \\ 0_{n,n} & 0_{n,n} & \cdots & 0_{n,n} & Q_0 & 0_{n,m} & 0_{n,m} & \cdots & 0_{n,m} \\ Q_{12}^T & 0_{m,n} & \cdots & 0_{m,n} & 0_{m,n} & Q_2 & 0_{m,m} & \cdots & 0_{m,m} \\ 0_{m,n} & Q_{12}^T & \cdots & 0_{m,n} & 0_{m,n} & 0_{m,m} & Q_2 & \cdots & 0_{m,m} \\ & & \ddots & & & & & \ddots & \\ 0_{m,n} & 0_{m,n} & \cdots & Q_{12}^T & 0_{m,n} & 0_{m,m} & 0_{m,m} & \cdots & Q_2 \end{bmatrix}.$$

The Problem 4.3 is identical to Problem 4.4 augmented with the additional constraint

$$v(k) = u^f(k)$$

where the vector $u^f(k) \in \mathbb{R}^b$ containing the b "free" elements of $u(k)$ (i.e., the elements of $u(k)$ whose indices i satisfy $\delta_i(k) = 1$) is arranged according to the increasing order of their indices. As a consequence, the optimal solutions of Problem 4.3 may be deduced from the optimal solutions of Problem 4.4 using the mapping (3.9).

The problem (4.18) is a mixed-integer quadratic program. It may be solved using many efficient academic and commercial solvers. In this monograph, this problem was solved using the solver CPLEX, whose MIP solver is based on the branch and bound method.

4.4.2 The Branch and Bound Method

The branch and bound method is a general search method for finding optimal solutions of discrete and combinatorial optimization problems. It was introduced in 1960 by *Land and Doig* [145] for the solving of the traveling salesman problem. This method aims at exploring, in an intelligent way, the space of feasible solutions of the problem. That's why it may be classified among the *implicit enumeration* methods. In the following, the principles of this algorithm will be described in the case of a minimization. In order to simplify our explanation, we will suppose that considered problem admits at least one optimal solution. Of course, the other cases may be easily taken into account.

4.4.2.1 General Concepts

As its name indicates it, this method is based on two complementary mechanisms: *branching* and *bounding*.

- Branching makes it possible to decompose a given problem into subproblems (by adding additional constraints), such that the union of the feasible solutions of these subproblems forms a partition (in the worst case a covering) of the feasible solutions of the original problem. In this manner, the resolution of the original problem is reduced to the resolution of the subproblems obtained by its branching. Branching induces a hierarchical relation between the different subproblems. This relation may be described and represented using concepts from the graph theory. In fact, in the branch and bound method, branching is applied in a recursive way to the subproblems where it may be possible (at a given stage of the execution of the algorithm), to find an optimal solution of the initial problem. As a result, the subproblems obtained by branching may be seen as the

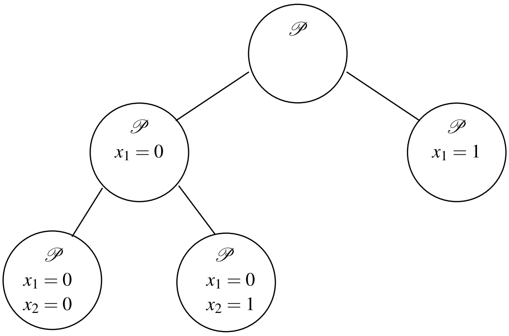
$\mathscr{P}$
$\mathscr{P}$ $x_1 = 0$
$\mathscr{P}$ $x_1 = 1$
$\mathscr{P}$ $x_1 = 0$ $x_2 = 0$
$\mathscr{P}$ $x_1 = 0$ $x_2 = 1$

Fig. 4.1 Search tree of problem $\mathscr{P}$

child nodes of the problem to which branching was applied, which is called *parent node*. Thus, all these nodes form a rooted tree, whose *root node* represents the initial problem to solve. This tree is usually called *search tree* or *decision tree*.

- Bounding consists in computing an upper and a lower bound of the optimal solution of a given node. The bounding stage allows the branch and bound to avoid exploring the nodes where it is possible to certify that they do not contain any optimal solution. In fact, if the upper bound of a subproblem **A** is larger than the lower bound of another subproblem **B**, then the optimal solution cannot lie in the feasible set of solutions of subproblem **A**. For that reason, it becomes useless to branch node **A**. Node **A** is then *pruned.*

A node is called *solved* if an optimal solution of the associated subproblem was obtained. This may occur, for example, when the constraints defining it (and which were added progressively along the different branching steps), reduce its set of feasible solutions to a singleton. In other situations, branching may make the subproblem sufficiently simple to be solved by polynomial algorithms. The solved nodes are the *final nodes* or *leave nodes* of the decision tree. The algorithm finishes when all the nodes were either solved or pruned.

Example 4.1 Figure 4.1 describes the search tree of problem $\mathscr{P}$ defined by

$$\mathscr{P}\begin{cases}\min\limits_{x} \quad 4x_1^2 - 3x_2^2 \\ \quad x_1 = 0, \\ \quad x_1, x_2 \in \{0, 1\}.\end{cases}$$

The branch and bound algorithm may be parameterized using the following four rules:

- *branching rules*, describing how to divide a given problem into subproblems,

- *bounding rules*, defining how to compute the upper and lower bounds of a given subproblem,
- *selection rules*, stating how to select the next problem to consider,
- *elimination rules*, describing how to recognize the subproblems that do not contain optimal solutions and that should be eliminated.

4.4.2.2 Application to Mixed-Integer Programming

To the best of the authors' knowledge, the application of the branch and bound method to solve mixed-integer nonlinear programs was proposed for the first time by *Dakin* [68]. The resolution of mixed-integer quadratic programs was studied by many authors, for example, [85, 86, 146]. In the paper by *Fletcher and Leyffer* [85], the branch and bound method was applied to solve mixed-integer quadratic programs, allowing to obtain better experimental results than the other methods.

In this paragraph, we consider programs in the form:

$$\mathscr{P}\begin{cases}\min\limits_{\mathscr{V}} \quad f(\mathscr{V}) = \dfrac{1}{2}\mathscr{V}^T\mathscr{F}\mathscr{V} + \mathscr{G}^T\mathscr{V} \\ \mathscr{A}\mathscr{V} \leq \mathscr{B}, \\ \mathscr{V}_i \in \{0,1\}, \quad \forall i \in I\end{cases}$$

where f is a positive semi-definite function and I is a set of indices indexing the Boolean components of $\mathscr{V}$. A basic branch and bound algorithm for solving program $\mathscr{P}$ is given in the following listing (Algorithm 4.1).

In Algorithm 4.1, U represents the cost of the best feasible solution of problem $\mathscr{P}$ at a given stage of the execution of the algorithm and L a lower bound of the optimal cost of $\mathscr{P}$. The set $\mathscr{M}$, used for the computation of L, represents the set of nodes that have been already evaluated (i.e., bounded) and that may contain the optimal solution. Problem $\mathscr{P}'$ denotes the problem obtained from $\mathscr{P}$ by relaxing the integrality constraints that are imposed on the variables indexed by I, that is, replacing the constraints

$$\mathscr{V}_i \in \{0,1\}, \quad \forall i \in I$$

by the constraints

$$\mathscr{V}_i \in [0,1], \quad \forall i \in I.$$

These variables are thus considered as continuous variables in $[0,1]$. Thus, $\mathscr{P}'$ is a quadratic program, and may be solved efficiently. Its solving is the most important part of the bounding or evaluation phase. In fact, the optimal cost of $\mathscr{P}'$ represents a lower bound of the optimal cost of $\mathscr{P}$.

Algorithm 4.1: Basic branch and bound algorithm

```
$\mathscr{L} := \{\mathscr{P}\}$;
$\mathscr{M} := \{\mathscr{P}\}$;
$U := +\infty$;
$L := -\infty$;
while $\mathscr{L} \neq \emptyset$ or $U - L > \varepsilon$ do
    select a node $\mathscr{Q}$ from list $\mathscr{L}$;
    remove the selected node $\mathscr{Q}$ from $\mathscr{L}$;
    solve the relaxed program $\mathscr{Q}'$ (bounding or evaluation of $\mathscr{Q}$);
    if $\mathscr{Q}'$ admits an optimal solution $x^*(\mathscr{Q}')$ of cost $J^*(\mathscr{Q}')$ and $J^*(\mathscr{Q}') < U$
    then
        lowerbound($\mathscr{Q}$) := $J^*(\mathscr{Q}')$;
        if $x^*(\mathscr{Q}')$ is integer-feasible then
            $U := J^*(\mathscr{Q}')$;
            $x^*(\mathscr{P}) := x^*(\mathscr{Q}')$;
            $J^*(\mathscr{P}) := J^*(\mathscr{Q}')$;
        else
            branch node $\mathscr{Q}$ and update list $\mathscr{L}$ with its child nodes;
        end if
        if all the brother nodes of $\mathscr{Q}$ were evaluated then
            remove the father node of $\mathscr{Q}$ from $\mathscr{M}$;
            put $\mathscr{Q}$ and its brother nodes in $\mathscr{M}$;
            $L = \min\{\text{lowerbound}(\mathscr{Q}), \mathscr{Q} \in \mathscr{M}\}$;
        end if
    else
        lowerbound($\mathscr{Q}$) := $+\infty$ (pruning of $\mathscr{Q}$);
    end if
end while
```

The algorithm begins by solving $\mathscr{P}'$ and giving its solution $x^*(\mathscr{P}')$. If this solution respects the integrality constraints, then it is also an optimal solution for $\mathscr{P}$ and the algorithm ends. Otherwise, there exists at least one non Boolean variable $\mathscr{V}_j$, $j \in I$ in $x^*(\mathscr{P}')$. The algorithm proceeds by branching problem $\mathscr{P}$ into two subproblems

$$\mathscr{Q}_1 \begin{cases} \min\limits_{\mathscr{V}} \quad \dfrac{1}{2}\mathscr{V}^T \mathscr{F} \mathscr{V} + \mathscr{G}^T \mathscr{V} \\ \mathscr{A}\mathscr{V} \leq \mathscr{B}, \\ \mathscr{V}_j = 0, \\ \mathscr{V}_i \in \{0, 1\}, \quad \forall i \in I_1 = I - \{j\} \end{cases}$$

and

$$\mathscr{Q}_2 \begin{cases} \min\limits_{\mathscr{V}} & \frac{1}{2}\mathscr{V}^T \mathscr{F} \mathscr{V} + \mathscr{G}^T \mathscr{V} \\ \mathscr{A}\mathscr{V} \leq \mathscr{B}, & \\ \mathscr{V}_j = 1, & \\ \mathscr{V}_i \in \{0, 1\}, & \forall i \in I_2 = I - \{j\} \end{cases}$$

that are added to the list $\mathscr{L}$. The procedure is reiterated by the selection of a subproblem $\mathscr{Q}$ from $\mathscr{L}$, based on the predefined selection rules. It may be possible that the relaxed subproblem $\mathscr{Q}'$ does not have any feasible solution. In this situation, it is pruned from the list $\mathscr{L}$ without being branched. The algorithm finishes when the list $\mathscr{L}$ becomes empty or when a suboptimal solution with a predefined absolute tolerance (defined by ε) is found.

4.4.3 An Illustrative Numerical Example

Consider the collection of three continuous-time linear time invariant (*LTI*) subsystems defined by

$$A_c^{(1)} = \begin{bmatrix} 0 & 130 \\ -800 & 10 \end{bmatrix}, \quad B_c^{(1)} = \begin{bmatrix} 0 \\ 224 \end{bmatrix}, \quad S^{(1)}$$

$$A_c^{(2)} = \begin{bmatrix} 0 & 14 \\ -250 & -200 \end{bmatrix}, \quad B_c^{(2)} = \begin{bmatrix} 0 \\ 620 \end{bmatrix}, \quad S^{(2)},$$

$$A_c^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 100 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & -10 & 0 \\ 11.6 & 0 & 1.184 & 0 \end{bmatrix}, \quad B_c^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 10.18 \end{bmatrix}, \quad S^{(3)}$$

where the subsystems $S^{(1)}$ and $S^{(3)}$ are open-loop unstable in opposition to the subsystem $S^{(2)}$ which is open-loop stable. The global system S composed by $S^{(1)}$, $S^{(2)}$ and $S^{(3)}$ may be described by the state matrix A_c and the input matrix B_c defined by

$$A_c = \mathrm{Diag}\big(A_c^{(1)}, A_c^{(2)}, A_c^{(3)}\big)$$

and

$$B_c = \mathrm{Diag}\big(B_c^{(1)}, B_c^{(2)}, B_c^{(3)}\big).$$

Assume that the global system S is controlled by a discrete-time controller executed at the sampling period $T_s = 2$ ms. The control commands are sent to the actuators through a bus that can carry at most one control command every 2 ms ($b = b_w = 1$). Assume now that the design criteria of the optimal continuous time controller for each subsystem are defined by matrices $Q_c^{(1)} = \mathrm{Diag}(100, 10)$, $R_c^{(1)} = 1$, $Q_c^{(2)} = \mathrm{Diag}(1000, 10)$, $R_c^{(2)} = 1$, $Q_c^{(3)} = \mathrm{Diag}(1000, 1000, 1, 1)$ and $R_c^{(3)} = 1$ and the desired closed-loop specifications for the global system are described by the closed-loop weighting matrices:

$$Q_c = \mathrm{Diag}\bigl(\mu_1 Q_c^{(1)}, \mu_2 Q_c^{(2)}, \mu_3 Q_c^{(3)}\bigr),$$
$$R_c = \mathrm{Diag}\bigl(\mu_1 R_c^{(1)}, \mu_2 R_c^{(2)}, \mu_3 R_c^{(3)}\bigr)$$

and

$$S_c = Q_c,$$

where μ_1, μ_2 and μ_3 are the weighting coefficients. In this example, μ_1, μ_2 and μ_3 were chosen equal to the inverse of steady state performance index of each separate subsystem controlled through a bus having an infinite bandwidth ($\mu_1 = 2.9$, $\mu_2 = 0.13$ and $\mu_3 = 0.016$).

Remark 4.1 The use of a resource-limited shared communication medium for the transmission of the control commands to the distributed actuators introduces a coupling between the three subsystems, which requires the weighting of relative importance of each subsystem using coefficients μ_1, μ_2 and μ_3. This contrasts with the optimal control problem without communication constraints, where these constants have no impact on the optimal control of the three independent subsystems.

The length of the optimal control and communication sequences is $N = 100$. An optimal solution with an error bound of 1×10^{-5} was required. The global system is started from the initial state $x(0) = [1\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T$. Its responses are depicted in Figs. 4.2 and 4.3. The optimal schedule is depicted in Fig. 4.4. In this schedule, $\delta_i = 1$ means that the bus is dedicated to the transmission of control signal u_i.

The first network slots are mainly dedicated to subsystem $S^{(1)}$ until it is stabilized. It may be observed that subsystem $S^{(2)}$, which is open-loop stable and whose response time is larger than $S^{(1)}$, needs only three time slots to be stabilized. After the stabilization of subsystems $S^{(1)}$ and $S^{(2)}$, the network resources are entirely dedicated to the stabilization of subsystem $S^{(3)}$. When the subsystem $S^{(3)}$ is close to the equilibrium state (from $t = 0.13$ s), then its control signals changes are minor. Consequently, the scheduling has no significant impact on control since the control signals of the three subsystems are relatively constant explaining thus the shape of the scheduling diagram, after $t = 0.13$ s. However, this optimal schedule is *dependent* on the *initial conditions*. If the initial condition $x(0)$ is modified, then the optimal control and schedule would be different. It is clear that such an open-loop schedule, which is generated off-line, cannot be applied at runtime as static schedule. In fact, assume that subsystem $S^{(1)}$ is disturbed at $t = 0.024$ s. Observing

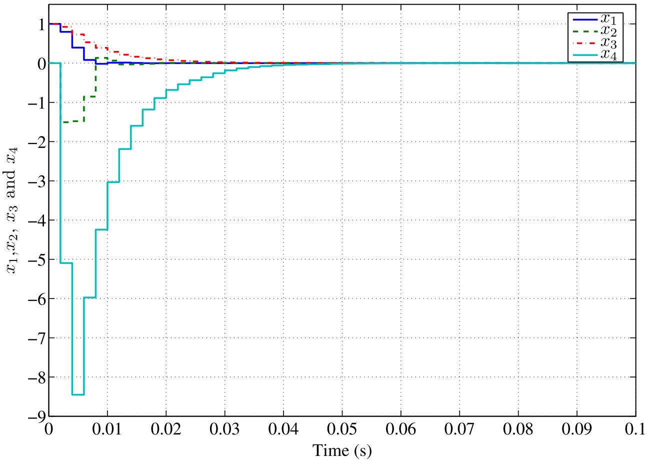

Fig. 4.2 Global system response—states x_1, x_2 (subsystem $S^{(1)}$) and x_3, x_4 (subsystem $S^{(2)}$) from $t = 0$ s to $t = 0.1$ s

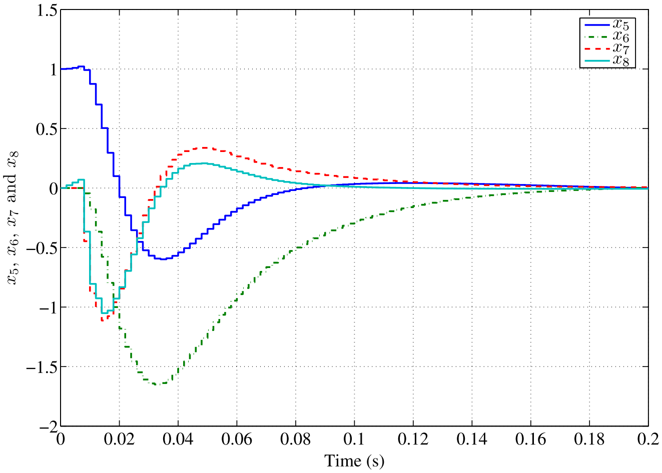

Fig. 4.3 Global system response—states x_5, x_6, x_7 and x_8 (subsystem $S^{(3)}$)

the schedule, no slots are allocated to the transmission of the messages of subsystem $S^{(1)}$ between $t = 0.024$ s and $t = 0.128$ s, which would induce performance degradations.

These observations show that in the same way as the optimal control, the optimal scheduling depends on the current state of the system. Consequently, fixed schedules may not be optimal if the system is started from another initial state or if it is disturbed at runtime.

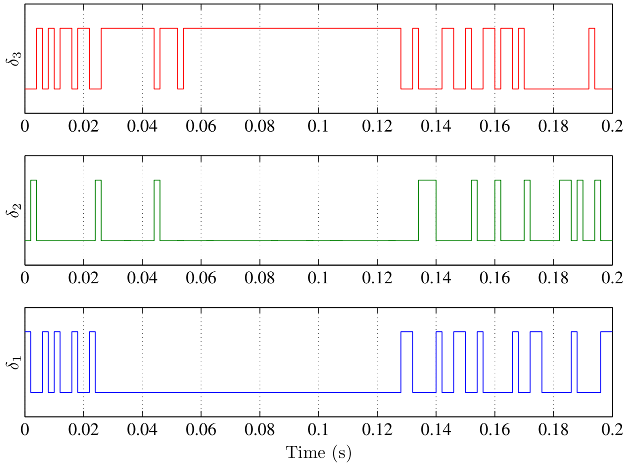

Fig. 4.4 Optimal scheduling of the controller-to-actuators link

4.5 Notes and Comments

In this chapter, we have started by studying the problem of optimal control for a given fixed finite communication sequence. Then, we have formulated and solved the problem of the joint optimization of control and scheduling, for a given initial state. The numerical examples illustrating this method have shown that the obtained optimal schedule depends on the chosen initial state $x(0)$. In other words, similarly to the standard optimal control case, the optimal scheduling depends on the current state of the system. However, from a computer science point of view, off-line scheduling has many advantages, essentially because it consumes a few computing resources and does not induce execution overheads. In order to obtain off-line schedules that are optimal from a certain point of view, it is necessary to use performance criteria that depend on the intrinsic characteristics of the system, not on a particular initial state. This will be the objective of the next chapter.

Nevertheless, this dependency between the optimal schedule and the plant state may be seen as a promising way for improving the quality of control by means of plant state-based scheduling algorithms. The design of such algorithms will be studied in Chap. 6.