Multi-layer Switching Control

Idin Karuei

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

August 2005

Your file Votre référence
ISBN: 0-494-10237-3
Our file Notre référence
ISBN: 0-494-10237-3

ABSTRACT

Multi-layer Switching Control

Idin Karuei

In this thesis, adaptive control of systems using switching techniques is investigated. It is assumed that the plant model belongs to a known finite set of models. It is also assumed that a set of controllers which solve the robust servomechanism problem for the family of plant models and a set of simultaneous stabilizers for certain subsets of plant models are given. It is shown that by using the above set of controllers and simultaneous stabilizers and choosing a proper switching sequence, one can minimize the number of switchings to destabilizing controllers. This can significantly improve the transient response of the system, which is one of the common weak points in most switching control schemes. Simulation results show the effectiveness of the proposed method in improving the transient response.

"When you look into the abyss,the abyss also looks into you." -Friedrich Nietzche

I dedicate this work to my mother and father my very first teachers that always encouraged me to discover this wonderful world.

I also dedicate this work to the imprisoned Iranian journalist Akbar Ganji, because of his fight for freedom of speech.

ACKNOWLEDGEMENTS

Thanks are due first to my supervisor, Dr. Amir G. Aghdam, for his help and encouragement. I also appreciate the help and support I received from my colleagues Mr. Shauheen Zahirazami and Mr. Mani Mesgarpour-Tousi for their useful comments and brilliant ideas for the multi-layer scheme.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

The control of a partially known plant has received considerable attention in the adaptive control literature. One of the relatively new lines of research in this area is switching control which was motivated to weaken the classical *a priori* information required in classical adaptive control and can be traced back to [1]. During the past several years, switching control schemes have been developed to accomplish a wide variety of tasks which would not have been possible using traditional adaptive control methods [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. On the other hand, when traditional adaptive control methods are ineffective due to large variations of system parameters, switching control can be applied to achieve the adaptive control objectives. Switching control is a combination of continuous dynamics with switching events which are logic-based [14]. Switching control methods have also been effectively used for fault recovery in control systems, when the set of possible failures is finite [15]. Another important application of switching control is in piece-wise affine and piece-wise linear approximation of nonlinear systems [16], [17], [18].

Switching control of systems using family of plant models was first introduced by Miller and Davison [4]. In this approach, it is assumed that the plant model always belongs to a finite set of known models called the family of plant models, or simply a family of plants. The changes of dynamics of the system are formulated by a sudden change from one of the plant models to another one in the set. Also a high-performance controller is designed for each plant in the family. When the plant changes, the current controller can no longer stabilize the system. There are different ways to observe this. In [19] and [8] for instance, performance indices are defined based on the identification error of the models. Their controllers, however, can be adaptive or fixed. The active controller is selected by using a hysteresis algorithm. The system will switch to a new controller if this new controller's performance index plus the hysteresis constant is less than the current controller's performance index. It is shown that adaptive models can be used in parallel to fixed models in order to achieve a good performance. In [4] some parameters related to each member of the family of plants and the closed-loop system corresponding to each plant model with its high-performance controller are defined. For each controller a so called "auxiliary signal" is built, which will then be compared to a filtered version of the output of the system. The switching controller operates in two phases: in phase 1 an upper bound on the magnitude of the initial state at any arbitrary time-instant is obtained. After phase 1 the system starts switching between each controller one-by-one and monitoring the corresponding auxiliary signal but at the same time some virtual auxiliary signals corresponding to other plant models whose controllers are to be examined one-by-one are also generated. If the norm of the output hits the auxiliary signal of the current controller, this controller is known to be destabilizing the unknown plant and as a result the system switches to another controller. Now it will monitor this controller's auxiliary signal and it will continue until it switches to a controller whose plant output magnitude does not meet the corresponding auxiliary signal. In this approach, there is no priority for any controller over the others. The auxiliary signal of a controller that is stabilizing the system is defined such that the absolute value of the output will never hit it which guaranties that after any change in the dynamics of the unknown plant, the algorithm will switch to each controller at most once. In contrast there are other works like [20], [21], and [22] which represent a scheme that may switch to each controller more than once. In this scheme, some generic assumptions such as observability and a known set of plant models are relaxed by allowing for cyclic switching at the expense of not guaranteeing that each controller is tried out only once. It will be seen later that this approach cannot be used in the proposed multi-layer switching control because in a multi-layer switching scheme it is essential to assume that if the plant switches from a controller to another one, that controller is either a destabilizing one, or a simultaneous stabilizer in the higher layers as will be discussed later.

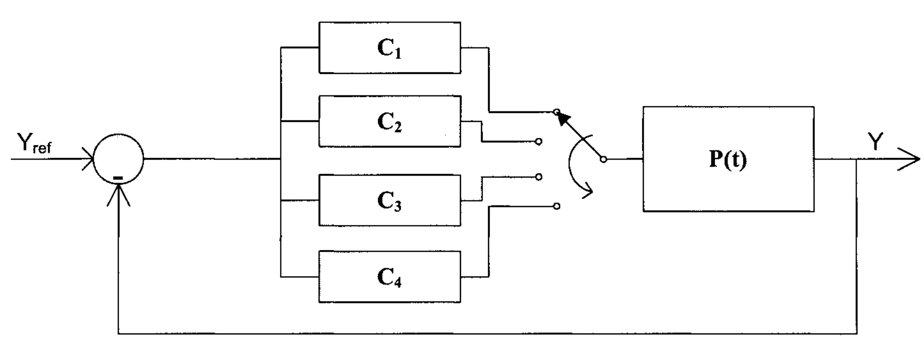

Figure 1.1: Switching control structure with a family of four controllers.

In [23], [24] and then in [12] and [25] a high level controller called "supervisor" and a family of controllers are introduced which solve the reference tracking problem of a plant with unmodeled dynamics. In this work the process is assumed to be SISO and linear. The transfer function is modeled by a union of a number of subclasses and each subclass is stabilized by one of the members of the family of controllers. The supervisor's role is to choose one of the controllers at each time instant to stabilize the system based on some logic. The controllers are selected by comparing normed-squared output estimations errors which are defined as performance signals. This switching control method can deal with parametric uncertainty, unmodeled dynamics, and exogenous disturbances which cannot be stabilized by linear feedback theory when the uncertainties are not sufficiently small. It is also shown that this method is very useful for uncertain non-minimum phase systems that cannot be stabilized by classical adaptive controller methods.

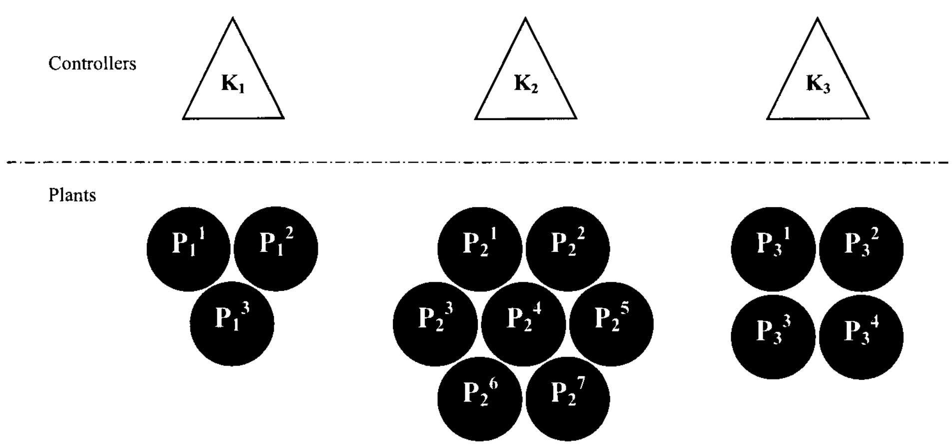

Figure 1.2: Morse's switching control with three controllers that can stabilize three subclasses of plants.

Switching control can be used for both discrete-time and continuous-time systems.

The application of computers and sampled-data systems has grown in practical control systems. This is due to the fact that computers and digital circuits can be easily programmed and used in different environments. Problems arise in the presence of unmodeled or changing dynamics. When classical adaptive control methods fail to stabilize such discrete-time systems, discrete-time switching control can potentially stabilize the system. In [26] a discrete-time supervisory control is given for tracking the reference input of a family of plants. The switching method of [12] is modified for discrete-time models in this paper. This method guarantees globally bounded states and zero-offset tracking.

By the late 80s a new trend in control of uncertain MIMO systems started as this type of systems have enormous real-world applications. For example, in [27], [28], [29], [30], [31], [32] the conventional adaptive control methods are extended to decentralized systems by making additional assumptions in the interconnection strength and the interconnection input channel while the *a priori* information for classical adaptive control of centralized systems are also required. In and [20] [22] the authors focus on generalizing the previous switching methods for MIMO (Multiple Input Multiple Output) plant models. Improvements in decentralized control motivated decentralized structures in switching control of MIMO systems [33]. In decentralized control of large-scale interconnected systems, it is desired to design a set of local controllers for the individual subsystems which solve the decentralized robust servomechanism problem for a given set of reference inputs and disturbance signals [34], [35], [36]. It is shown that when changes in dynamics of the system or uncertainty are too large to be stabilizable by a fixed decentralized controller, switching control methods can be very effective [13], [33]. In this approach the plant is assumed to belong to a so called "family of plants" and a set of decentralized controllers is designed for each plant model. A switching mechanism is then used to switch between the candidate controllers to stabilize the system and achieve the desired reference-tracking and disturbance rejection objectives. Decentralized control systems have a more complex structure than the centralized counterparts. The pseudo-decentralized switching method proposed in [13] uses a set of upper bound signals for different control agents, which are to be compared to the system outputs. This method uses a modified version of the switching mechanism given in [4]. Switching occurs when all control agents hit their corresponding upper bound signals. As a result, a weak link between the control agents is required. The decentralized switching control proposed in [33], on the other hand, requires a set of stabilizing local control agents, i.e., the set of possible plant models need not be known.

As discussed earlier, switching control can be considered as an alternative to classical adaptive control methods with the following pros and cons. One of the advantages of switching control compared to the conventional adaptive control is its effectiveness for stabilizing non-minimum phase systems. Secondly, switching control design is, in general, simpler than the conventional adaptive control counterparts because the design procedure has only two steps: designing a set of candidate controllers which solve the robust servomechanism problem [11], [20] for all possible plant models, and then applying a proper switching mechanism. Furthermore, switching control can be applied to the system with unknown high-frequency gain sign and also unknown relative degree [37], [38]. More importantly, switching methods are very useful for highly uncertain systems which are known to be difficult to control by applying traditional adaptive techniques. Performance of the switching controller depends on each individual controller as well as the switching rule. Moreover, in presence of any error in polarity of the input or output signal and also any type of fault occurrence, switching control methods are proven efficient [15].

Therefore, switching control can also be very effective in fault recovery problems.

However, the main disadvantage of switching control is its bad transient response in general. Several methods have been proposed to tackle this shortcoming of switching control [19], [39]. One of the main reasons for undesirable transient response is that in the transition from the initial controller to the final one, the system may switch to several destabilizing controllers. This is due to the fact that the system may switch to several non-matching controllers and each switching to a controller which fails to stabilize the system would cause an undesirable overshoot in the output signal. This gives the motivation to the present work to improve the transient performance by reducing the number of switchings to unstable controllers so that the method can be widely used in practical applications.

In this thesis, a method is proposed to improve the transient response of the switching control systems by reducing the number of switchings to destabilizing controllers. The proposed method utilizes different layers of controllers with different properties. This is an extension of the switching method introduced in [4], which assumes that the set of plant models $\{\mathbf{P}_i : i = 1, 2, \ldots, p\}$ is given and upper-bounds on the disturbance and reference input magnitudes are available. It is also assumed that each plant is controllable and observable. The plant may change slowly and the changes of the plant are unknown. When a change in dynamics of the plant or any other parameter like polarity occurs, the switching system intelligently chooses an ordered subset of the set of all controllers, switches to each of them one by one and waits for a finite time on each controller until stability or instability is sensed. If the system is known to be unstable at the most recent switched controller the best matching controller is determined and will be switched to immediately. It is to be noted that the proposed multi-layer architecture can be applied to any switching mechanism that does not switch more than once to each controller. In other words, one can use switching control methods other than the one given in [4] as long as it has the above mentioned property.

Switching in the system occurs when the norm of the error signal becomes greater than or equal to the corresponding upper-bound signal. In this proposed multi-layer scheme, $p-2$ layers of controllers are designed, where p denotes the number of models in the family of plants. Layer $k \in \{2,\ldots,p-2\}$ consists of a set of controllers which have the property that each one stabilizes k plants in the family and destabilizes the remaining $p-k$ plants. Layer 1 consists of a set of p controllers, where each one solves the robust servomechanism problem for one of the models in the family. The main difference between the previous switching methods and the proposed one is that the additional controllers which represent layers $2,\ldots,p-2$ are used to improve the transient response. Throughout this thesis, the previous switching control methods will be referred to as single-layer switching.

The idea is based on the information gained after each switching event. In single-layer switching methods the unknown plant is a member of a set and each time the system is known to be unstable with a controller, only one of the possible plant models will be taken out of that set and the set will become smaller by each switching to an unstable controller. By this definition any single-layer switching mechanism that does not switch to a controller more than once is a special case of multi-layer switching with an incomplete structure. In multi-layer switching some of the controllers can stabilize more than one plant. As a result, if one of such controllers destabilizes the system, more than one plant is taken out of the set. By choosing a smart switching scheme the number of switchings to unstable controllers will be decreased. It is shown in this work that this number is at most one when certain number of controllers of the multi-layer structure exist. It is also shown that the number of unstable switchings is less than or equal to that of single-layer methods for the cases when a subset of the required controllers for complete multi-layer structure exists.

This thesis is organized as follows. At first, the single-layer methods are shown which are the starting point for this work in Chapter 2. A detailed survey on the single-layer switching methods that are the basis of this thesis is included with a numerical example to show the pros and cons of switching control. Chapter 3 and Chapter 4 contain most of the contributions of this work and include new theoretical results. A simple ideal method of multi-layer switching together with some examples is given in Chapter 3 whose effectiveness is illustrated through a numerical example that compares it to the single-layer method. In Chapter 4 a more general method is given that can be used for systems with a large number of plants and controllers or a set of controllers in different layers that do not have the ideal structure. The formulation given in this chapter makes multi-layer and/or single-layer switching control more applicable in practice. Conclusions and future work are discussed in Chapter 5.

Chapter 2

Single-layer Switching Control

Classical adaptive control tehcniques are widely used to stabilize systems with uncertain dynamics. For systems with abrupt parameter changes, however, traditional adaptive control methods are usually ineffective. Such abrupt changes can be due to fast change of the environment under which the system is operating. They can also be due to fault occurrence in the system. Furthermore, many adaptive techniques fail to control non-minimum phase systems, systems with unknown high frequency gain sign, or systems with unknown relative degree. Switching control is an alternative to classical adaptive control that can solve this type of problem more efficiently. In switching control, it is often assumed that the plant model belongs to a known set of models, e.g. see [4]. Any possible change in dynamics of the system is characterized by a model in the set. A family of controllers is then designed with the property that each controller can stabilize one and only one of the plant models in the given set. Changes in dynamics of the plant may happen at anytime and once they occur, the current controller can no longer stabilize the plant. The system should now switch to another controller. This method of switching is only dependant on the reference signals, output signals, and the inputs to the plant which are the outputs of the active controller. p boundary signals which are also referred to as "auxiliary signals" are generated from the outputs, reference inputs, and the outputs of the controller which are related to p different models in the set. The output of the system, or a filtered version of it (for avoiding the discontinuity) is compared to the auxiliary signal related to the active controller while all other auxiliary signals are generated in parallel. If the output of the system meets the boundary which is the auxiliary signal of the current controller, that controller is known not to be the stabilizing one that the system will switch to another controller. The method presented in [4] guarantees that each controller is tried out at most once. They have proven that the auxiliary signal is hit by the filtered output if and only if the controller does not stabilize the plant. Since each controller can only stabilize one of the plant models, the system will eventually switch to the correct matching controller. Throughout this thesis a switching scheme with a set of controllers that have one-to-one correspondence with the models in the family will be referred to as single-layer. A Multi-layer switching, on the other hand, has a control structure that consists of the single-layer controllers as well as a set of simultaneous stabilizers as will be discussed later. In this chapter the single-layer switching scheme of [4] is presented, which will be compared to the multi-layer switching in Chapter 3.

2.1 Switching Control Structure

The switching control configuration consists of several controllers which are designed to meet some specifications. For a plant characterized by a set of p plant models, a set of p controllers is also required. Each controller can be adaptive or fixed, continuous-time or discrete-time, and should stabilize one and only one of the plant models. The switching mechanism given in section 2.4 is particularly given for LTI controllers.

The controller is made of two parts: the set of non-switching controllers, and the supervisor which performs the switching mechanism. For a switching system with no controller preference such as the one given in [4], the supervisor is very simple. It should switch to each controller one by one in any arbitrary order. Note that the model indices in a single-layer structure are assigned arbitrarily, for simplicity and without loss of generality, the supervisor is assumed to switch in an ascending order until it reaches the last controller and then it rolls back to the first one.

2.2 Problem Formulation

It is assumed that the current plant $\mathbf{P}(t)$ belongs to a known finite set of plant models given by

$$\forall t : \mathbf{P}(t) \in \Pi = \{\mathbf{P}_i : i \in \bar{\mathrm{p}}\} \tag{2.1}$$

$$\bar{\mathrm{p}} = \{1, 2, \ldots, p\} \tag{2.2}$$

It is also assumed that each plant model in the above set is described by the following state-space equations

$$\dot{x} = A_i x + B_i u + E_i w \tag{2.3a}$$

$$y = C_i x + F_i w \tag{2.3b}$$

$$e = y_{ref} - y \tag{2.3c}$$

where $i \in \bar{\mathrm{p}}$ and $x(t) \in \mathbb{R}^{n_i}$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^r$ is the output, $w(t) \in \mathbb{R}^v$ is the disturbance signal and $e(t) \in \mathbb{R}^r$ is the error.

It is assumed that for each $i \in \bar{\mathrm{p}}$ there exists a high performance controller $\mathbf{K}_i$ of the form

$$\dot{z} = G_i z + H_i u + J_i y_{ref} \tag{2.4a}$$

$$u = K_i z + L_i y + M_i y_{ref} \tag{2.4b}$$

This set represents the family of controllers and is denoted by Φ.

The index of each controller represents the plant that can be stabilized by that controller, e.g. $\mathbf{K}_i$ stabilizes plant model $\mathbf{P}_i$, and destabilizes the other plants in the set.

The closed-loop control law corresponding to the controller $\mathbf{K}_i$ can be written in the following form [4]

$$\tilde{u} = \tilde{K}_i \tilde{y} \tag{2.5}$$

which results in a stable system corresponding to the controllable and observable plant $\mathbf{P}_i$ as follows

$$\dot{\tilde{x}} = \tilde{A}_i \tilde{x} + \tilde{B}_i \tilde{u} + \tilde{E}_i w \tag{2.6a}$$

$$\tilde{y} = \tilde{C}_i \tilde{x} + \tilde{D}_i y_{ref} + \tilde{F}_i w \tag{2.6b}$$

where

$$\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} u \\ \dot{z} \end{bmatrix}, \quad \tilde{y} = \begin{bmatrix} y \\ z \\ y_{ref} \end{bmatrix} \tag{2.7}$$

and

$$\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, \tilde{B}_i = \begin{bmatrix} B_i & 0 \\ 0 & I \end{bmatrix}, \tilde{C}_i = \begin{bmatrix} C_i & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix},$$

$$\tilde{D}_i = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}, \tilde{E}_i = \begin{bmatrix} E_i \\ 0 \end{bmatrix}, \tilde{F}_i = \begin{bmatrix} F_j \\ 0 \\ 0 \end{bmatrix},$$

$$\tilde{K}_i = \begin{bmatrix} L_i & K_i & M_i \\ H_i & G_i & J_i \end{bmatrix}$$

Definition 1 *Throughout this paper, a switching to a destabilizing controller will be called an unstable switching.*

Figure 2.1 shows an architecture of eight controllers for eight plant models where the plant models are represented by black circles and controllers are represented by triangles.

2.3 Single-layer Switching Algorithm

According to the above formulation, the controllers have no preference over each other therefore the switching method is very simple. The controllers are put in a desired order and will be picked one-by-one when the system is known to be unstable. Instability is observed when the norm of the outputs of the system hits a boundary function.

Assume that the plant is stabilized by controller i_1. Once a change in model occurs, the new plant model is known to be one of the remaining $p-1$ models $\bar{\mathrm{p}} - \{i_1\} = \{i_2, i_3, \ldots, i_p\}$, which results in instability of the closed-loop system. The following assumption is made for the development of the proposed algorithm.

Assumption 1 *Each plant can be stabilized by one and only one controller.*

It is to be noted that the models are meant to be chosen far enough from each other so that each controller can stabilize only one of the models in most cases.

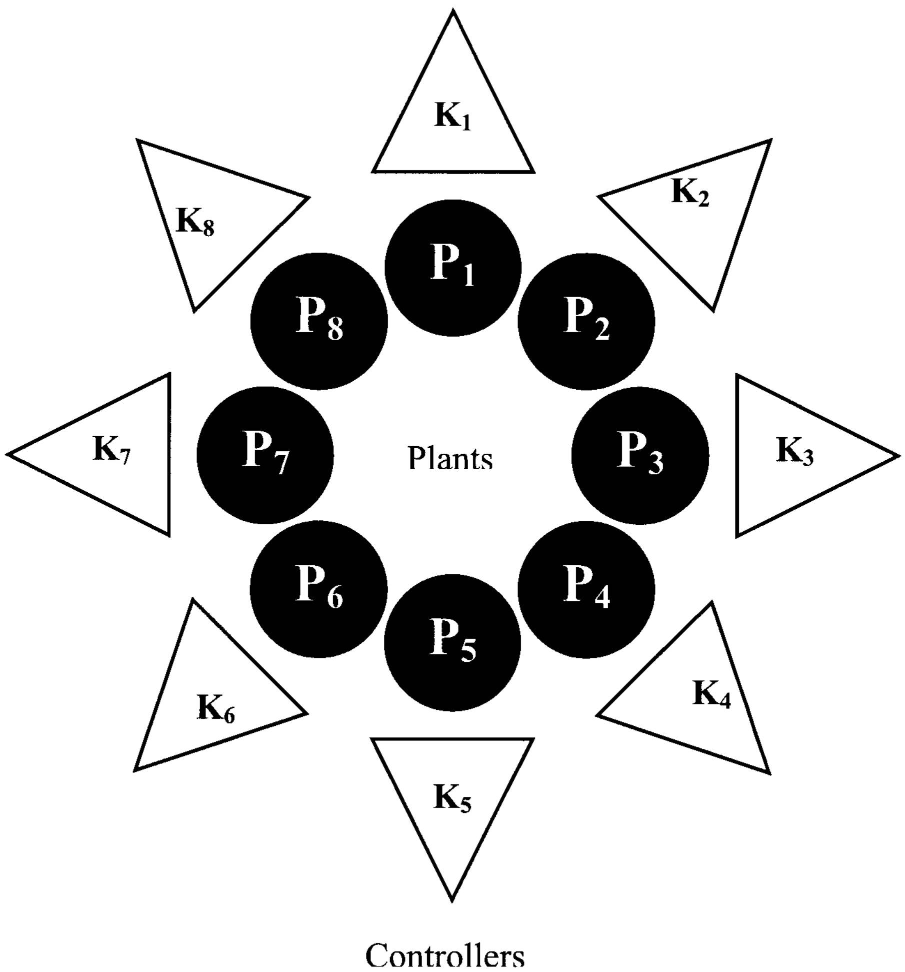

Figure 2.1: A single layer switching structure for eight plant models.

Algorithm 1

1) *Set i the same as the index of the current controller. If the system becomes unstable at controller* $\boldsymbol{K}_i$ *go to next step else repeat* 1.

2) *If* $i = p$ *set* $i = 1$ *else set* $i = i + 1$.

3) *Switch to controller* $\boldsymbol{K}_i$ *and go to step* 1.

Example 1 *Assume that there is a family of 8 plants as shown in Figure 2.2. Initially, the actual plant model is* $\boldsymbol{P}_7$ *which is stabilized by controller* $\boldsymbol{K}_7$*. Assume that the plant model changes to* $\boldsymbol{P}_4$ *at time* t_0 *which is the new unknown plant. The system becomes unstable and it switches to* $\boldsymbol{K}_8$*. The system becomes unstable and switches to* $\boldsymbol{K}_1$ *and becomes unstable again and then goes to* $\boldsymbol{K}_2$ *which causes again instability of the closed-loop system. After another switching to a destabilizing controller* $\boldsymbol{K}_3$ *the system eventually switches to the correct controller which is* $\boldsymbol{K}_4$.

2.4 Switching Mechanism

The switching instants will be obtained by using the same approach as in [4]. The method consists of two phases. Because the initial states of the system cannot be known and are affecting the whole system, at first, a bound on the initial condition is obtained which is possible and then an upper-bound function is introduced using the information gained in the previous phase which will be compared to the norm of the output of the system or a filtered version of that. If the norm of the output of the system meets the upper-bound, the current controller is known to be destabilizing the system and it should be switched to another controller. The desired controller is found by continuing switching between different controllers.

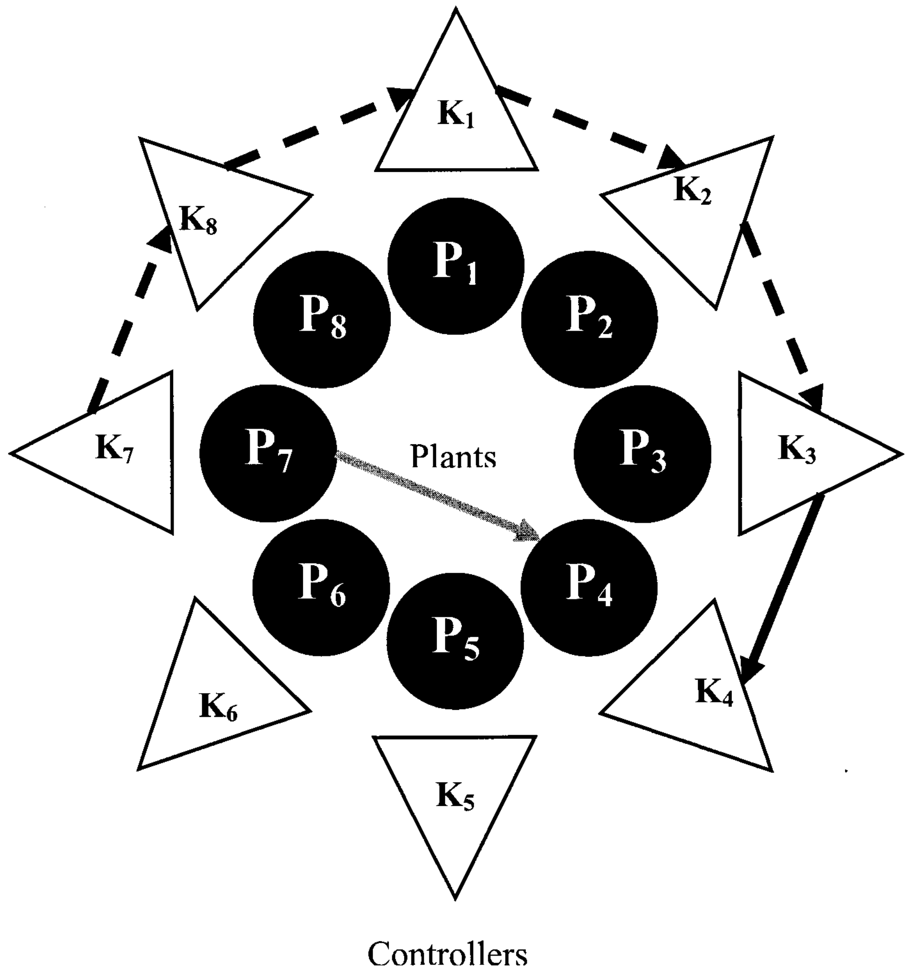

Figure 2.2: Single layer switching. Dark dashed arrows represent unstable switchings and dark solid arrow denotes a stable switching.

Note that there are p auxiliary signals for a family of p plant models. The signal related to the plant that can be stabilized by the current controller is chosen as the upper-bound while other auxiliary signals are also generated at the same time for next switchings.

2.4.1 Finding a Bound on the Initial Condition

The following result from Lemma 1 of [4] given in **Appendix A** provides an upper-bound for the initial condition with $u(.) = 0$

$$\|x(0)\|^2 \leq \alpha_{i_1} \int_0^T \|y(\tau)\|^2 d\tau + \alpha_{i_2}\tilde{b} \tag{2.8}$$

where

$$\begin{aligned}
W_i &= \int_0^T e^{A_i'\tau} C_i' C_i e^{A_i\tau} d\tau, \\
\alpha_{i_3} &= \text{the smallest singular value of } W_i, \\
\alpha_{i_1} &= 2/\alpha_{i_3}, \\
\alpha_{i_2} &= (2/\alpha_{i_3}) \int_0^T \left[\ \textstyle\int_0^t \|C_i e^{A_i(t-\tau)} E_i\| d\tau + \|F_i\| \ \right]^2 dt \\
\tilde{b} &= \text{upper-bound on disturbance.}
\end{aligned}$$

With $\tilde{u}(t) = 0$ for $t \in [0, T]$ and T is an arbitrary time chosen for the first phase, and $z(0) = 0$, find

$$\theta := \int_0^T \|y(\tau)\|^2 d\tau \tag{2.10}$$

and define the following ϕ upper-bound signals for all stable closed-loop configurations:

$$\begin{aligned}
\dot{r}_i(t) &= \lambda_i r_i(t) + \gamma_{i_2} \|\tilde{K}_i(\tilde{y} - \tilde{D}_i y_{ref})\| + \gamma_{i_3}\tilde{b}\ , \quad t \in [0, T] \\
r_i(0) &= 0
\end{aligned} \tag{2.11}$$

where from Lemma 2 of [4] given in **Appendix A** there exist $\lambda_i < 0$ and $\gamma_{i_1} > 0$ such that

$$\|e^{(\tilde{A}_i+\tilde{B}_i\tilde{K}_i\tilde{C}_i)t}\| \leq \gamma_{i_1} e^{\lambda_i t} \tag{2.12}$$

and

$$\gamma_{i_2} = \gamma_{i_1} \|\tilde{B}_i\| \tag{2.13a}$$

$$\gamma_{i_3} = \gamma_{i_1} \|\tilde{E}_i + \tilde{B}_i\tilde{K}_i\tilde{F}_i\| \tag{2.13b}$$

Define

$$\mu_i = [\alpha_{i_1}\theta + \alpha_{i_2}\tilde{b}^2]^{\frac{1}{2}}$$

Assuming that $\|w(t)\| \leq \tilde{b}$, if the actual plant model is $\mathbf{P}_i$, it follows from (2.8) that $\|x(0)\| \leq \mu_i$

2.4.2 Searching for the Correct Controller

In this phase, control action is applied and the upper-bound signal introduced in [4] is given by

$$\dot{r}_i(t) = \lambda_i r_i(t) + \gamma_{i_2}\|\tilde{u}(t) - \tilde{K}_i(\tilde{y}(t) - \tilde{D}_i y_{ref})\| + \gamma_{i_3}\tilde{b} \tag{2.14}$$

with initial condition

$$r_i(T^+) = r_i(T) + \gamma_{i_1} e^{\lambda_i r_i T} \mu_i \tag{2.15}$$

Each closed-loop controller-plant pair has an upper-bound signal which is a function of the norm of the error. It is often desired to use a smooth error signal by applying a filter as follows

$$\dot{\tilde{r}} = \tilde{\lambda}\tilde{r}(t) + (\lambda - \tilde{\lambda})\|\tilde{y}(t) - \tilde{D}y_{ref}\|, \;\; \tilde{r}(T) = 0 \tag{2.16}$$

where $\tilde{\lambda} < \min\{\lambda_i : i \in p\}$.

Each time the filtered error signal meets the upper-bound signal corresponding to the current controller $\mathbf{K}_i$ instability is detected. In other words, the system will switch to another candidate controller when

$$\tilde{r}(t) = \|\tilde{C}_i\| r_i(t) + \|\tilde{F}_i\| \tilde{b} + \varepsilon \;, \tag{2.17}$$

and ε is an arbitrary positive value [4].

Theorem 1 *Using the switching sequence of Algorithm 1, and the switching instants t_i which represent the times that the filtered signal meets the upper-bound corresponding to the i^{th} switched controller with $t_0 := 0$, the system will eventually switch to the correct controller and never switches twice to any of the controllers.*

Proof of Theorem 1: The proof follows immediately from the results of Theorem 1 in [4].

2.5 Numerical Examples

An example is given in this section to clarify the idea of switching and show its effectiveness and weaknesses.

Example 2 *Consider the following unstable non-minimum phase plant:*

$$\boldsymbol{P} = \lambda \frac{s-1}{(s-2)(s+1)}, \quad 1 < \lambda(t) < 6$$

A family of four plant models $\boldsymbol{P}_i = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4\}$ is then considered as follows

$$\boldsymbol{P}_1 = \frac{s-1}{(s-2)(s+1)}, \; \boldsymbol{P}_2 = 2\boldsymbol{P}_1, \; \boldsymbol{P}_3 = 4\boldsymbol{P}_1, \; \boldsymbol{P}_4 = 6\boldsymbol{P}_1$$

The high-performance controllers of first layer are obtained as follows

$$
\begin{aligned}
\boldsymbol{K}_1 &= \frac{448s^2+450s-18}{31s(s-9)}, \\
\boldsymbol{K}_2 &= \frac{1}{2}\times\frac{448s^2+450s-18}{31s(s-9)}, \\
\boldsymbol{K}_3 &= \frac{1}{4}\times\frac{448s^2+450s-18}{31s(s-9)}, \\
\boldsymbol{K}_4 &= \frac{1}{6}\times\frac{448s^2+450s-18}{31s(s-9)}.
\end{aligned}
$$

Assume that initially the actual plant model is $\boldsymbol{P}_1$ *and at some point of time it changes to* $\boldsymbol{P}_4$*. When it becomes unstable, the system will switch from* $\boldsymbol{K}_1$ *to* $\boldsymbol{K}_2$*, then to* $\boldsymbol{K}_3$*, and finally to* $\boldsymbol{K}_4$*. The first two switching instants are unstable. Now assume that the plant model changes again from* $\boldsymbol{P}_4$ *to* $\boldsymbol{P}_3$*. The switching method switches from* $\boldsymbol{K}_4$ *to* $\boldsymbol{K}_1$*, then to* $\boldsymbol{K}_2$*, and finally to* $\boldsymbol{K}_3$*. Two unstable switchings occur using the single-layer method. Figures 2.3 and 2.4 show the high magnitude of the transient response due to switching to destabilizing controllers.*

In this section a practical example is given to clarify the method.

Example 3 *The plant is a mass-spring-damper structure as shown in Figure 2.5 .*

It is a linear system which can be represented by the following state-space model:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

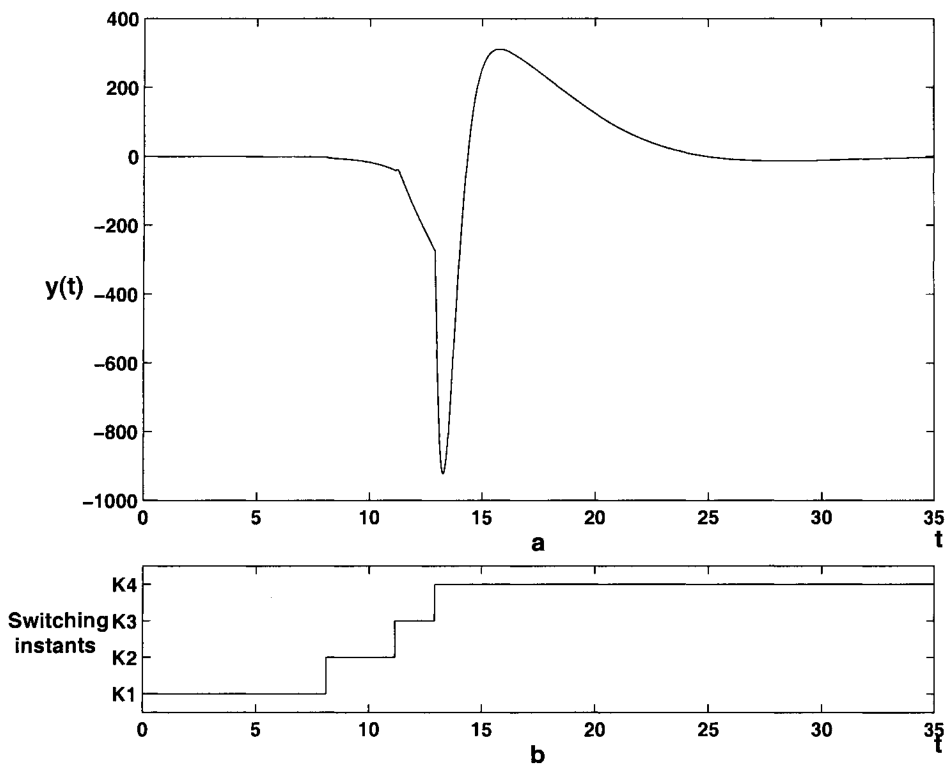

Figure 2.3: Closed-loop simulation results for Example 2, using the single-layer scheme, when the plant model changes from P_1 to P_4. (a) Output signal; (b) switching instants.

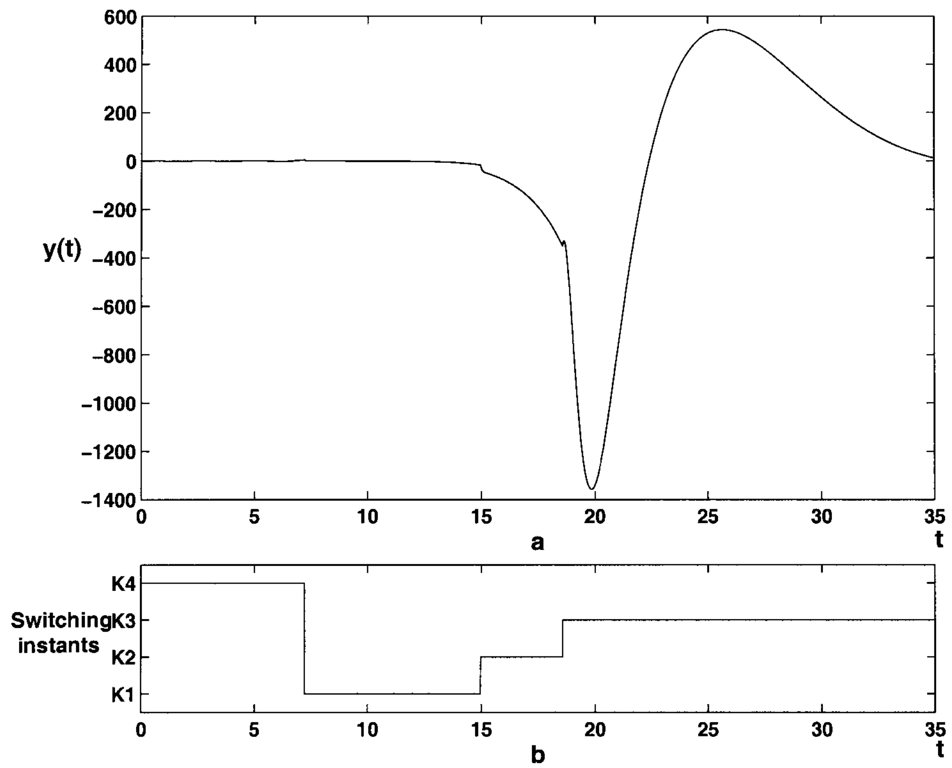

Figure 2.4: Closed-loop simulation results for Example 2, using the single-layer scheme, when the plant model changes from P_4 to P_3. (a) Output signal; (b) switching instants.

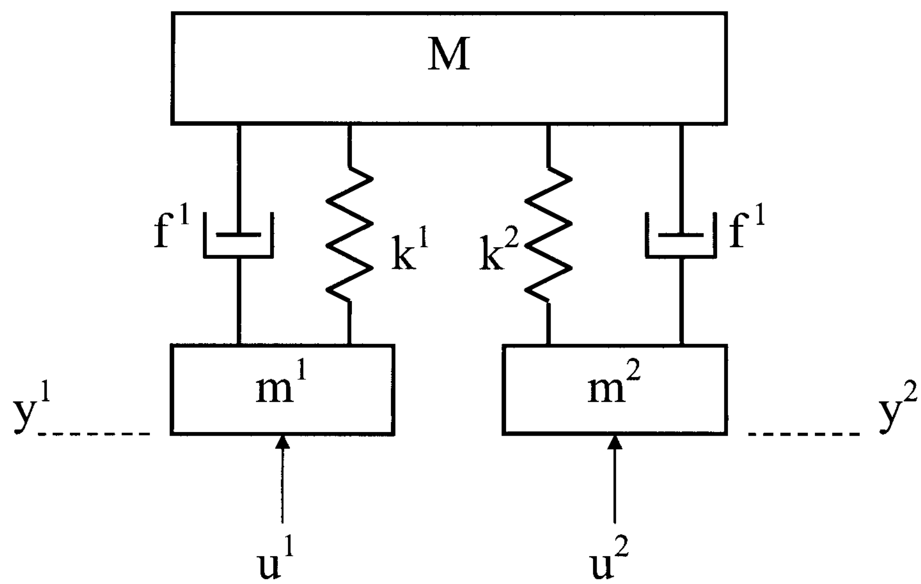

Figure 2.5: Mass-Spring-Damper system of Example3

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k^1+k^2}{M} & -\frac{f^1+f^2}{M} & \frac{k^1}{M} & \frac{f^1}{M} & \frac{k^2}{M} & \frac{f^2}{M} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k^1}{m^1} & \frac{f^1}{m^1} & -\frac{k^1}{m^1} & -\frac{f^1}{m^1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{k^2}{m^2} & \frac{f^2}{m^2} & 0 & 0 & -\frac{k^2}{m^2} & -\frac{f^2}{m^2} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b^1 & 0 \\ 0 & 0 \\ 0 & b^2 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The changing parameters are as follows: M, m^1, and m^2 are the masses, k^1 and k^2 are the spring coefficients, f^1 and f^2 are the damper coefficients, and b^1 and b^2 are the polarity of the inputs which can be 1 *or* -1.

The family of the plants contains 4 *plant models:*

P$_1$: $M_1 = 1.7363$, $m_1^1 = 0.1874$, $m_1^2 = 0.1333$, $k_1^1 = 0.1897$, $k_1^2 = 0.1751$, $f_1^1 = 0.2787$, $f_1^2 = 0.5716$, $b_1^1 = b_1^2 = 1$

P$_2$: $M_2 = 0.1692$, $m_2^1 = 1.7363$, $m_2^2 = 1.3011$, $k_2^1 = 0.1603$, $k_2^2 = 0.9085$, $f_2^1 = 0.5873$, $f_2^2 = 0.4125$, $b_2^1 = b_2^2 = 1$

P$_3$: $M_3 = 1.7363$, $m_3^1 = 0.1874$, $m_3^2 = 0.1333$, $k_3^1 = 0.1897$, $k_3^2 = 0.1751$, $f_3^1 = 0.2787$, $f_3^2 = 0.5716$, $b_3^1 = b_3^2 = -1$

P$_4$: $M_4 = 0.1692$, $m_4^1 = 1.7363$, $m_4^2 = 1.3011$, $k_4^1 = 0.1603$, $k_4^2 = 0.9085$, $f_4^1 = 0.5873$, $f_4^2 = 0.4125$, $b_4^1 = b_4^2 = -1$

Four controllers are enough for the above family of plants. These controllers can be described by state-space models, where u is the input vector, e is the error signal, and z is *the state of the controller. The controllers are designed using continuous linear-quadratic-gaussian control synthesis (LQG) in MATLAB. These are optimal dynamic controllers and are as follows:*

Controller 1:

$$S_1 = \begin{bmatrix} 0 & 0 & -0.9945 & -0.1049 & -1.231 & -2.588 & -1.748 & -1.249 & -0.3228 & -0.226 \\ 0 & 0 & 0.1049 & -0.9945 & -0.9134 & -3.924 & 0.02395 & -0.226 & -2.263 & -0.8544 \\ 0 & 0 & -2.054 & -0.517 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.517 & -2.065 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1.562 & -1.461 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5019 & -0.6663 & -0.2101 & -0.4897 & 0.1093 & 0.1605 & 0.1009 & 0.3292 \\ 0 & 0 & -1.743 & -1.115 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1.488 & -0.9841 & -0.2191 & -1.101 & -2.76 & -2.736 & -0.3228 & -0.226 \\ 0 & 0 & -1.015 & -1.766 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -0.5686 & -1.574 & 0.4005 & 0.3642 & 0.02395 & -0.226 & -3.576 & -5.143 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2.054 & 0.517 \\ 0.517 & 2.065 \\ 1.562 & 1.461 \\ 0.5019 & 0.6663 \\ 1.743 & 1.115 \\ 0.4938 & 0.8792 \\ 1.015 & 1.766 \\ 0.6735 & 0.5796 \end{bmatrix}, Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, K_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Controller 2:

$$S_2 = \begin{bmatrix} 0 & 0 & -0.9984 & -0.05587 & 0.1081 & -0.1393 & -2.387 & -2.266 & -0.3979 & -0.2724 \\ 0 & 0 & 0.05587 & -0.9984 & -0.3737 & -0.1911 & 0.006612 & -0.2724 & -2.182 & -2.197 \\ 0 & 0 & -2.201 & -0.2079 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.2079 & -2.154 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.7635 & -1.404 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.4199 & -0.6978 & -6.317 & -5.909 & 0.9476 & 3.471 & 5.37 & 2.438 \\ 0 & 0 & -1.943 & -0.4467 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1.563 & -0.4765 & 0.2004 & 0.199 & -2.48 & -2.604 & -0.3979 & -0.2724 \\ 0 & 0 & -0.4584 & -1.841 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -0.3904 & -1.453 & 0.3245 & 0.1259 & 0.006612 & -0.2724 & -2.88 & -2.514 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2.201 & 0.2079 \\ 0.2079 & 2.154 \\ 0.7635 & 1.404 \\ 0.4199 & 0.6978 \\ 1.943 & 0.4467 \\ 0.5644 & 0.4207 \\ 0.4584 & 1.841 \\ 0.4462 & 0.4547 \end{bmatrix}, Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, K_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Controller 3:

$$S_3 = S_1\ ,\ R_3 = R_1\ ,\ Q_3 = -Q_1\ ,\ K_3 = -K_1$$

Controller 4:

$$S_4 = S_2\ ,\ R_4 = R_2\ ,\ Q_4 = -Q_2\ ,\ K_4 = -K_2$$

Assume that initially the plant model is $\boldsymbol{P}_1$ and the controller is $\boldsymbol{K}_1$. The plant model suddenly changes to $\boldsymbol{P}_3$ so that the system becomes unstable. After the upper-bound is hit, the algorithm switches to $\boldsymbol{k}_2$ and then goes to $\boldsymbol{K}_3$. The transient response can be seen in Figure 2.6.

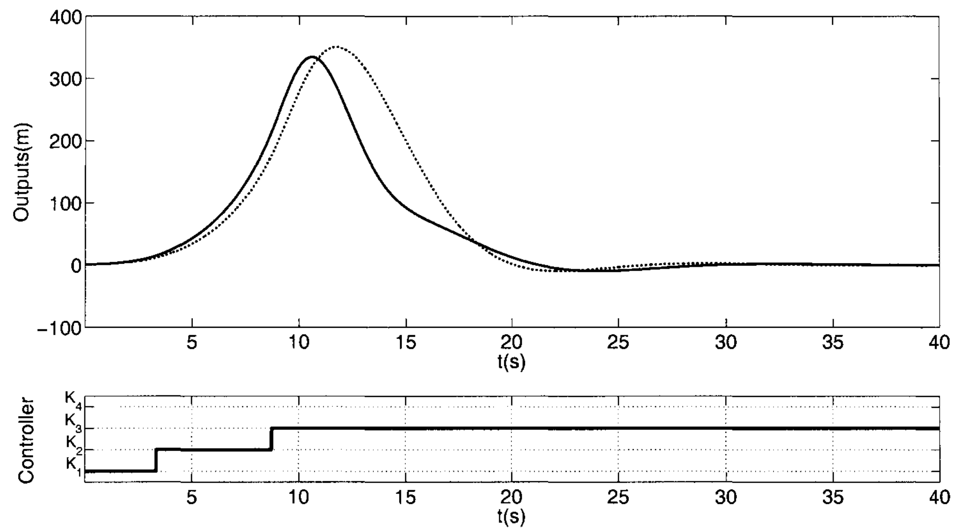

Figure 2.6: Closed-loop simulation results for Example 3, using the single-layer scheme, when the plant model changes from P_1 to P_3. (a) Output signal; (b) Switching instants.

Chapter 3

Multi-layer Switching Control

One of the problems of switching control methods is the bad transient response which is mainly contributed by unguided switching to destabilizing controllers before the system locks onto the correct controller. When the plant dynamics change from one model to another one, the system may switch to some other destabilizing controllers until it finds the correct controller. Addition of other layers of controllers can potentially reduce the number of "unstable switchings".

The first layer is just the same as the set of controllers as in a single-layer method. The second layer contains controllers that can stabilize two and only two of the plant models and so on the n^{th} layer is a set of controllers that can stabilize n and only n members of the family of plants. It can easily be seen that there can be at most p layers of controllers for a family of p plant models. It will be shown later that layer p which is made of only one controller is useless in all cases because switching to this controller does not make any changes while it adds some time to the transient response and a non-necessary stable switching which does not gain any information about the system. Layer $p-1$ is not necessary in most of the cases according to the algorithm given in the following parts which means that the structure is made of $p-2$ layers of controllers in almost all cases.

The controllers on the first layer are chosen by solving a desired optimality problem because there is no condition for them except stabilizing their related plant. It is to be noted that single-layer structure is a special case of multi-layer structure. It will be shown later that another condition in single-layer methods can be relaxed in a multi-layer algorithm. In single-layer each controller should only stabilize one of the plants which may result in sub-optimal controllers in some cases when the optimal controller for a plant can stabilize another plant. The problem rises when the supervisor switches to a controller that is not the matching controller for the plant but is stabilizing it. In advanced multi-layer algorithm however, these cases are taken into account in a way that the designer just needs to design optimal controllers. The algorithm works in a way that controllers in the first layer that can stabilize more than one plant are used in a way that the supervisor does not get trapped.

The initial point of the system is considered to be a controller in the first layer because they are chosen to get the best performance that is defined for the system. When the plant changes, the system becomes unstable. There are several other controllers that may stabilize or destabilize the plant. None of these controllers are preferred to others in a single-layer algorithm which is based on stability of the plants, but a multi-layer algorithm can choose the controllers wisely so that the number of unstable switchings is decreased and the transient response is improved as a result.

In this chapter a multi-layer structure is introduced and a simple algorithm of switching is given that will be shown to be an acceptable solution to improving the bad transient response of the so called "single-layer switching" methods through some examples.

3.1 Problem Formulation

It is assumed that the current plant $\mathbf{P}(t)$ belongs to a known finite set of plant models given by

$$\forall t : \mathbf{P}(t) \in \Pi = \{\mathbf{P}_i : i = 1, 2, \ldots p\} \tag{3.1}$$

$$\bar{\mathrm{p}} = \{1, 2, \ldots, p\} \tag{3.2}$$

It is also assumed that each plant model in the above set is described by the following state-space equations

$$\dot{x} = A_i x + B_i u + E_i w \tag{3.3a}$$

$$y = C_i x + F_i w \tag{3.3b}$$

$$e = y_{ref} - y \tag{3.3c}$$

where $i \in \bar{\mathrm{p}}$, $x(t) \in \mathbb{R}^{n_i}$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^r$ is the output, $w(t) \in \mathbb{R}^v$ is the disturbance signal and $e(t) \in \mathbb{R}^r$ is the error.

As in the previous works, it is assumed that for each $i \in \bar{\mathrm{p}}$ there exists a high performance controller $\mathbf{K}_i$ of the form

$$\dot{z} = G_i z + H_i u + J_i y_{ref} \tag{3.4a}$$

$$u = K_i z + L_i y + M_i y_{ref} \tag{3.4b}$$

This set represents the first layer of controllers in our proposed multi-layer architecture and is denoted by Φ_1

$$\Phi_1 = \{\mathbf{K}_i : i \in \bar{\mathrm{p}}\}, \quad N(\Phi_1) = N(p) \tag{3.5}$$

where $N(.)$ gives the number of the elements of a set. On the other hand, the set of controllers of layer $k, k = 2, \ldots, p$ is denoted by Φ_k, as follows

$$\Phi_k = \{\mathbf{K}_{i_1 i_2 \ldots i_k} | i_1, i_2, \ldots, i_k \in \bar{\mathrm{p}}\} \tag{3.6}$$

where $i_j, j = 1, \ldots, k$ are distinct integers and the indices of each controller represent the plants that can be stabilized by that controller, e.g. $\mathbf{K}_{i_1 i_2 \ldots i_k}$ "only" stabilizes plant models $\mathbf{P}_{i_1}, \mathbf{P}_{i_2}, \ldots, \mathbf{P}_{i_k}$, and destabilizes the other plants in the set.

According to the above definition in each layer $k \in \bar{p}$, there exist $N(\Phi_k) \times k$ combinations of stable closed-loop configurations, e.g. each controller on layer k can stabilize k plant models and therefore has k stable closed-loop configuration which should be then multiplied by the number of the controllers on that layer. The total number of all stable configurations corresponding to all controller layers is given by

$$\phi = \sum_{k=1}^{p-2} N(\Phi_k) \times k$$

The closed-loop control law corresponding to the controller $\mathbf{K}_{i_1 i_2 \ldots i_k}$ can be written in the following form [4]

$$\tilde{u} = \tilde{K}_{i_1 i_2 \ldots i_k} \tilde{y} \tag{3.7}$$

which results in a stable system corresponding to the controllable and observable plant $\mathbf{P}_j : j \in \{i_1, i_2, \ldots i_k\}$ as follows

$$\dot{\tilde{x}} = \tilde{A}_j \tilde{x} + \tilde{B}_j \tilde{u} + \tilde{E}_j w \tag{3.8a}$$

$$\tilde{y} = \tilde{C}_j \tilde{x} + \tilde{D}_j y_{ref} + \tilde{F}_j w \tag{3.8b}$$

where

$$\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} u \\ \dot{z} \end{bmatrix}, \quad \tilde{y} = \begin{bmatrix} y \\ z \\ y_{ref} \end{bmatrix} \tag{3.9}$$

and

$$\tilde{A}_j = \begin{bmatrix} A_j & 0 \\ 0 & 0 \end{bmatrix}, \tilde{B}_j = \begin{bmatrix} B_j & 0 \\ 0 & I \end{bmatrix}, \tilde{C}_j = \begin{bmatrix} C_j & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix},$$

$$\tilde{D}_j = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}, \tilde{E}_j = \begin{bmatrix} E_j \\ 0 \end{bmatrix}, \tilde{F}_j = \begin{bmatrix} F_j \\ 0 \\ 0 \end{bmatrix},$$

$$\tilde{K}_{i_1 i_2 \ldots i_k} = \begin{bmatrix} L_{i_1 i_2 \ldots i_k} & K_{i_1 i_2 \ldots i_k} & M_{i_1 i_2 \ldots i_k} \\ H_{i_1 i_2 \ldots i_k} & G_{i_1 i_2 \ldots i_k} & J_{i_1 i_2 \ldots i_k} \end{bmatrix}$$

According to the above formulation, there are different layers of controllers which are used in the multi-layer switching method. It is desired now to find a switching path which consists of at most one unstable switching in general, between the controllers of different layers.

Figure 3.1 shows an architecture of controller layers for 6 plant models where the plant models are represented by black circles and controllers of layer 1,2,3 and 4 are represented by triangles, squares, pentagons and hexagons, respectively.

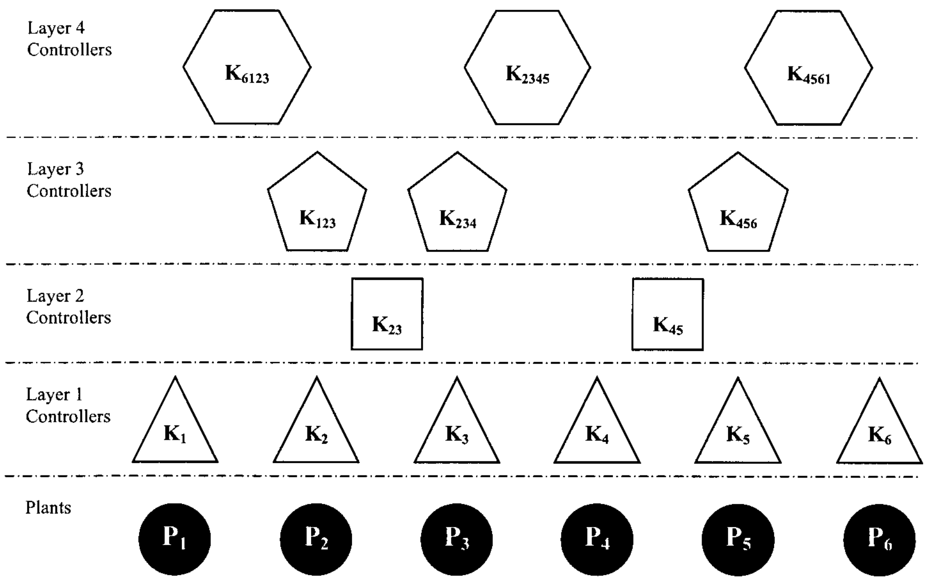

Figure 3.1: Four layers of controllers for six plant models.

3.2 Multi-layer Switching Algorithm

Unlike the set of controllers in a single-layer structure that are the same in the case that each one can stabilize only one plant, in a multi-layer structure the controllers are at different sets which categorize them according to the number of plants they can stabilize. It results in more switching options that can be confusing but, using the following definitions, a simple algorithm is given that guaranties the number of unstable switchings to be less than or equal one.

Definition 2 *Any controller whose indices include all but one of the indices of another controller is called a "parent" of that controller. For instance,* $\boldsymbol{K}_{i_1 i_2 \ldots i_{k-1}}$ *is a parent of* $\boldsymbol{K}_{i_1 i_2 \ldots i_k}$*. On the other hand, any controller in a layer other than layer one is a "child" to its parent in the lower layer.*

Definition 3 *Any two controllers in the same layer which have only one uncommon index are called "sister" of each other, i.e. all but one of the plant models that can be stabilized by a controller, can also be stabilized by its sister in that layer. It is to be noted that in a multi-layer switching structure, sister controllers potentially have a unique common parent.*

Definition 4 *A "child-parent switching route" is a switching path from a controller in one of the higher layers to a controller in the first layer that consists of only child-parent controllers.*

Definition 5 *A structure that is made of all possible controllers in different layers is called an ideal structure. Notice that an ideal structure has $p-1$ layers and $\sum_{i=1}^{p-1} \frac{p!}{(p-i)!i!} = 2^p - 2$ controllers.*

Assumption 2 *i) There exists a controller in layer $p-2$ which destabilizes two of plant models and stabilizes all other models.*
ii) Each of the plant models is destabilized by at least one controller in layer $p-2$.
iii) There exists a child-parent route from any of the controllers in layer $p-2$ to at least one controller in the first layer.

Algorithm 2

1) *If the system became unstable at controller $\boldsymbol{K}_{i_1}$ set $l = 1$ and $\bar{I} := \{i_2, i_3, \ldots, i_p\}$. It is known that the plant model belongs to the set $\{\boldsymbol{P}_i | i \in \bar{I}\}$.*

2) *Switch to a controller in layer $p - (l + 1)$ which destabilizes $\boldsymbol{P}_{i_1}$. The other plant model which is also destabilized by this controller is denoted by $\boldsymbol{P}_{i_{k_l}}$. If the closed-loop system becomes unstable, the actual plant model is identified to be $\boldsymbol{P}_{i_{k_l}}$. Switch to $\boldsymbol{K}_{i_{k_l}}$ and*

stop. Otherwise, set $\bar{I} := \bar{I} - \{i_{k_l}\}$ *and* $l := l + 1$. *Let the current controller be denoted by* $\boldsymbol{K}_{\bar{I}}$. *Go to step* 3.

3) *If* $l = p - 1$ *stop. Otherwise go to step* 4.

4) *Switch to a parent controller of* $\boldsymbol{K}_{\bar{I}}$ *in layer* $p - (l + 1)$. *The plant model which was stabilized by the previous controller but is destabilized by this controller is denoted by* $\boldsymbol{P}_{i_{k_l}}$. *If the closed-loop system becomes unstable, the actual plant model is identified to be* $\boldsymbol{P}_{i_{k_l}}$. *Switch to* $\boldsymbol{K}_{i_{k_l}}$ *and stop. Otherwise, set* $\bar{I} := \bar{I} - \{i_{k_l}\}$ *and* $l := l + 1$. *Let the current controller be denoted by* $\boldsymbol{K}_{\bar{I}}$. *Go to step* 3.

It can be verified that using the switching sequence described in Algorithm 2, it is guaranteed that the system will eventually switch to the correct controller with at most one unstable switching, provided all required controllers in different layers exist. This is due to the fact that on the first switching instant or any child-to-parent switching the system either becomes stable or unstable and because there is only one plant that can be stabilized by the child but can not be stabilized by the parent, instability causes the plant to be known and the next switching would be the last switching which results in stability. In other words, when a controller can stabilize the system the algorithm switches to its parent and if the parent controller destabilizes the system the first unstable switching has happened and also the plant model is known to be the plant with the uncommon index of the child controller and the parent controller. Therefore it would be the last unstable switching as well and the next switching is a stable switching to the matching controller on the first layer.

Example 4 *Assume that there is a family of 6 plants as shown in Figure 3.2. Initially, the actual plant model is* $\boldsymbol{P}_4$ *which is stabilized by controller* $\boldsymbol{K}_4$. *Assume that the plant* *model changes to* $\boldsymbol{P}_2$ *at time* t_0 *which is the new unknown plant. The system becomes unstable and it switches to* $\boldsymbol{K}_{6123}$ *which stabilizes* $\boldsymbol{P}_1$*,* $\boldsymbol{P}_2$*,* $\boldsymbol{P}_3$*,* $\boldsymbol{P}_6$ *and destabilizes* $\boldsymbol{P}_4$ *and* $\boldsymbol{P}_5$*. The system becomes stable and switches to* $\boldsymbol{K}_{123}$ *which is the parent of the previous controller* $\boldsymbol{K}_{6123}$*. The system remains stable and should switch to a parent of the current controller. The current controller has three parents* $\boldsymbol{K}_{12}$*,* $\boldsymbol{K}_{13}$*, and* $\boldsymbol{K}_{23}$*. Assume that the system switches to* $\boldsymbol{K}_{23}$*. The system becomes stable and should switch to one of the parents of* $\boldsymbol{K}_{23}$*. Assume that it switches to* $\boldsymbol{K}_3$*. This controller destabilizes the system and it is the only time the system becomes unstable. At this point, the new actual plant model is identified to be* $\boldsymbol{P}_2$ *and the system switches to* $\boldsymbol{K}_2$*.*

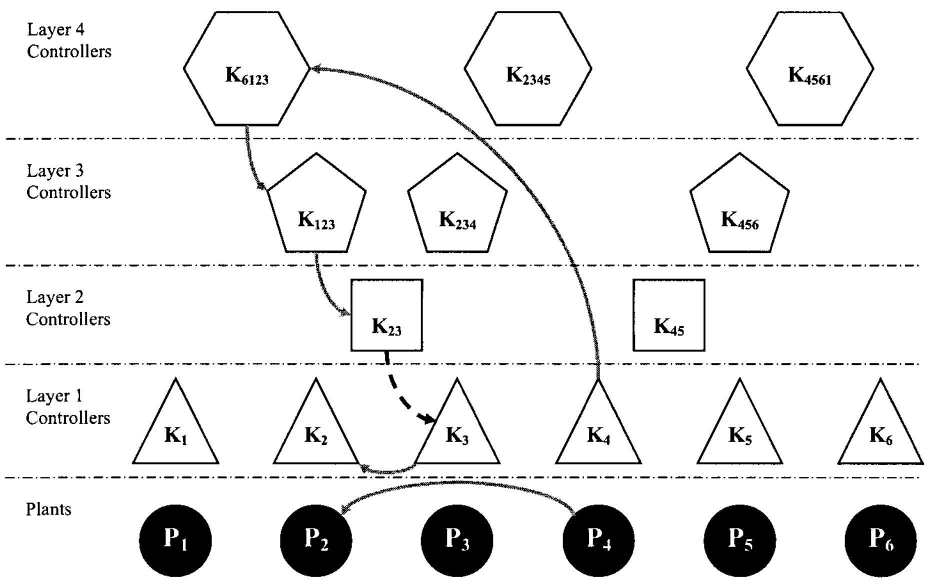

Figure 3.2: Switching in four layers. Solid arrows represent stable switchings and dashed arrow denote an unstable switching.

3.3 Structure of Layers

A multi-layer structure can contain several controllers for a family of plants. It is not necessary to have all these controllers in order to reduce the number of unstable switchings to less than or equal one. Assume that the conditions of Assumption 2 are met. Then, once the plant model changes within the known set of plant models, it is guaranteed that the system will switch to the correct controller after at most one unstable switching.

Remark 1 *It is to be noted that the number of all possible controllers for layer k that can stabilize k plant models and destabilize the remaining $p-k$ models is equal to $\frac{p!}{k!(p-k)!}$. Thus, the total number of all possible controllers in the proposed multi-layer structure is equal to*

$$\sum_{k=1}^{p-2} \frac{p!}{k!(p-k)!}$$

More specifically, the number of all possible controllers for layer $p-2$ is equal to $\frac{p(p-1)}{2}$. However, it can be easily verified that only $fix(\frac{p+1}{2})$ controllers for layer $p-2$ would suffice, where $fix(.)$ represents the nearest integer towards zero. Similarly, the maximum number of required controllers for layers $2,3,\ldots,p-2$ is equal to $p-2$. Since designing simultaneous stabilizers for layers $2,3,\ldots,p-2$ can be difficult in practice, one may design at most $p-2$ required controllers for each layer.

Example 5 *Assume that a family of seven plants $\Pi = \{\boldsymbol{P}_1,\boldsymbol{P}_2,\boldsymbol{P}_3,\boldsymbol{P}_4,\boldsymbol{P}_5,$ $\boldsymbol{P}_6,\boldsymbol{P}_7\}$ is given. The proposed multi-layer architecture consists of five layers. There can be $\frac{7\times 6}{2} = 21$ controllers on the highest layer theoretically but only 4 of them are necessary.*

The conditions of Assumption 2 can be satisfied in different ways. For instance, one can choose the set of controllers $\{\boldsymbol{K}_{12345}, \boldsymbol{K}_{34567}, \boldsymbol{K}_{56712}, \boldsymbol{K}_{71234}\}$ *to represent* Φ_5*. The next layers should then be designed in a way that child-parent routes exist from all controllers of* Φ_5 *to those of layer 1.* $\Phi_4 = \{\boldsymbol{K}_{2345}, \boldsymbol{K}_{3456}, \boldsymbol{K}_{6712}, \boldsymbol{K}_{7123}\}$ *is one of the several choices for layer four. Since pair* $\boldsymbol{K}_{2345}$ *and* $\boldsymbol{K}_{3456}$ *and pair* $\boldsymbol{K}_{6712}$ *and* $\boldsymbol{K}_{7123}$ *have common parent, a smart choice for controllers of the third layer would be* $\Phi_3 = \{\boldsymbol{K}_{345}, \boldsymbol{K}_{712}\}$*. Layer two can then be chosen as* $\Phi_2 = \{\boldsymbol{K}_{34}, \boldsymbol{K}_{71}\}$*. Finally, the first layer should have seven controllers as in a single-layer architecture.*

3.4 Switching Mechanism

The switching instants will be obtained by using the same approach as in [4]. The method consists of two phases. First, a bound on the initial condition is obtained and then the desired controller is found by switching between different controllers.

3.4.1 Finding a Bound on the Initial Condition

The following result from Lemma 1 of [4] given in **Appendix A** provides an upper-bound for the initial condition with $u(.) = 0$

$$\|x(0)\|^2 \leq \alpha_{i_1} \int_0^T \|y(\tau)\|^2 d\tau + \alpha_{i_2} \tilde{b} \tag{3.10}$$

where T is a desired time that phase 1 lasts and

$$W_i = \int_0^T e^{A_i'\tau} C_i' C_i e^{A_i \tau} d\tau,$$

$$\alpha_{i_3} = \text{the smallest singular value of } W_i,$$

$$\alpha_{i_1} = 2/\alpha_{i_3},$$

$$\alpha_{i_2} = (2/\alpha_{i_3}) \int_0^T \left[\int_0^t \|C_i e^{A_i(t-\tau)} E_i\| d\tau + \|F_i\| \right]^2 dt$$

$$\tilde{b} = \text{upper-bound on disturbance.}$$

With $\tilde{u}(t) = 0$ for $t \in [0,T]$ and $z(0) = 0$, find

$$\theta := \int_0^T \|y(\tau)\|^2 d\tau \tag{3.12}$$

and define the following ϕ upper-bound signals for all stable closed-loop configurations:

$$\begin{aligned} \dot{r}_{j,i_1 i_2 \dots i_k}(t) &= \lambda_j r_{j,i_1 i_2 \dots i_k}(t) + \gamma_{(j,i_1 i_2 \dots i_k)_2} \|\tilde{K}_{i_1 i_2 \dots, i_k}(\tilde{y} - \tilde{D}_j y_{ref})\| \\ &\quad + \gamma_{(j,i_1 i_2 \dots i_k)_3} \tilde{b} \ , \ \ t \in [0,T] \\ r_{j,i_1 i_2 \dots i_k}(0) &= 0 \end{aligned} \tag{3.13}$$

where from Lemma 2 of [4] given in **Appendix A** there exist $\lambda_{j,i_1 i_2 \dots i_k} < 0$ and $\gamma_{(j,i_1 i_2 \dots i_k)_1} > 0$ such that

$$\|e^{(\tilde{A}_j + \tilde{B}_j \tilde{K}_{i_1 i_2 \dots i_k} \tilde{C}_j)t}\| \leq \gamma_{(j,i_1 i_2 \dots i_k)_1} e^{\lambda_{j,i_1 i_2 \dots i_k} t} \tag{3.14}$$

and

$$\gamma_{(j,i_1 i_2 \dots i_k)_2} = \gamma_{(j,i_1 i_2 \dots i_k)_1} \|\tilde{B}_j\| \tag{3.15a}$$

$$\gamma_{(j,i_1 i_2 \dots i_k)_3} = \gamma_{(j,i_1 i_2 \dots i_k)_1} \|\tilde{E}_j + \tilde{B}_j \tilde{K}_{i_1 i_2 \dots i_k} \tilde{F}_j\| \tag{3.15b}$$

Define

$$\mu_j = [\alpha_{j_1}\theta + \alpha_{j_2}\tilde{b}^2]^{\frac{1}{2}}$$

Assuming that $\|w(t)\| \leq \tilde{b}$ if the actual plant model is $\mathbf{P}_j$, it follows from (3.10) that $\|x(0)\| \leq \mu_j$

3.4.2 Searching For The Correct Controller

In this phase, control action is applied and the upper-bound signal introduced in [4] is given by

$$\begin{aligned} \dot{r}_{j,i_1 i_2 \ldots i_1}(t) &= \lambda_j r_{j,i_1 i_2 \ldots i_k}(t) + \gamma_{(j,i_1 i_2 \ldots i_k)_2}\|\tilde{u}(t) - \tilde{K}_{i_1 i_2 \ldots, i_k}(\tilde{y}(t) - \tilde{D}_j y_{ref})\| \\ &\quad + \gamma_{(j,i_1 i_2 \ldots i_k)_3}\tilde{b} \end{aligned} \tag{3.16}$$

with initial condition

$$r_{j,i_1 i_2 \ldots i_k}(T^+) = r_{j,i_1 i_2 \ldots i_k}(T) + \gamma_{(j,i_1 i_2 \ldots i_k)_1} e^{\lambda_j r_{j,i_1 i_2 \ldots i_k} T}\mu_j \tag{3.17}$$

Each closed-loop controller-plant pair has an upper-bound signal which is a function of the norm of the error. It is often desired to use a smooth error signal by applying a filter as follows

$$\dot{\tilde{r}} = \tilde{\lambda}\tilde{r}(t) + (\lambda - \tilde{\lambda})\|\tilde{y}(t) - \tilde{D}y_{ref}\|, \;\; \tilde{r}(T) = 0 \tag{3.18}$$

where $\tilde{\lambda} < \min\{\lambda_i : i \in p\}$.

Each time the filtered error signal meets the upper-bound signal corresponding to the current controller $\mathbf{K}_{i_1 i_2 \ldots i_k}$ and plant $\mathbf{P}_j$ ($j \in \{i_1, i_2, \ldots, i_k\}$), instability is detected. In other words, the system will switch to another candidate controller when

$$\tilde{r}(t) = \|\tilde{C}_j\| r_{j,i_1 i_2 \ldots i_k}(t) + \|\tilde{F}_j\| \tilde{b} + \varepsilon \ , \tag{3.19}$$

and ε is an arbitrary positive value [4].

The switching sequence of MULTI LAYER ALGORITHM requires that the system switches from the higher layer controllers to the lower layer ones even if the system is stabilized in a higher layer. Unlike unstable switchings, stable switchings cannot be identified through the upper-bound signals. In order to detect stability, a sufficiently long time-interval will be used such that if the norm of error does not meet the upper-bound signal, the system is stable. This time-interval can be obtained by considering worse case scenario associated with initial conditions, reference input and disturbance signal. It can also be obtained experimentally. This time duration will be called maximum time for stability and will be denoted by t_d.

Lemma 1 *The controller $\boldsymbol{K}_{i_1 i_2 \ldots i_k}$ destabilizes the system iff the filtered error signal meets any of the upper-bound signals corresponding to $\boldsymbol{P}_{i_1}, \boldsymbol{P}_{i_2}, \ldots \boldsymbol{P}_{i_k}$ and controller $\boldsymbol{K}_{i_1 i_2 \ldots i_k}$.*

Proof of Lemma 1: Suppose that the current controller is $\mathbf{K}_{i_1 i_2 \ldots i_k}$ which can stabilize both $\mathbf{P}_{i_1}$ and $\mathbf{P}_{i_2}$. For simplicity and with no loss of generality, it will be assume that $\tilde{C}_i : i \in p$ has a unit norm. The upper-bound signals $r_{i_1,i_1 i_2 \ldots i_k}$ and $r_{i_2,i_1 i_2 \ldots i_k}$ are used as in [4] and are obtained from (3.13) and (3.16). Two new upper-bound signals are defined as follows

$$\begin{aligned} \dot{r}'_{i_1,i_1 i_2 \ldots i_k}(t) &= \lambda_{i_1 i_2 \ldots i_k} r'_{i_1,i_1 i_2 \ldots i_k}(t) \\ &+ \gamma_{(i_1 i_2 \ldots i_k)_2} \|\tilde{u}(t) - \tilde{K}_{i_1 i_2 \ldots i_k}(\tilde{y}(t) - \tilde{D}_j y_{ref}(t))\| + \gamma_{(i_1 i_2 \ldots i_k)_3} \bar{b} \end{aligned} \tag{3.20}$$

$$\dot{r}'_{i_2,i_1i_2\ldots i_k}(t) = \lambda_{i_1i_2\ldots i_k} r'_{i_2,i_1i_2\ldots i_k}(t)$$
$$+ \gamma_{(i_1i_2\ldots i_k)_2} \|\tilde{u}(t) - \tilde{K}_{i_1i_2\ldots i_k}(\tilde{y}(t) - \tilde{D}_j y_{ref}(t))\| + \gamma_{(i_1i_2\ldots i_k)_3} \bar{b} \tag{3.21}$$

where

$$\gamma_{(i_1i_2\ldots i_k)_1} = \max(\gamma_{(i_1,i_1i_2\ldots i_k)_1}, \gamma_{(i_2,i_1i_2\ldots i_k)_1}), \tag{3.22a}$$

$$\gamma_{(i_1i_2\ldots i_k)_2} = \max(\gamma_{(i_1,i_1i_2\ldots i_k)_2}, \gamma_{(i_2,i_1i_2\ldots i_k)_2}), \tag{3.22b}$$

$$\gamma_{(i_1i_2\ldots i_k)_3} = \max(\gamma_{(i_1,i_1i_2\ldots i_k)_3}, \gamma_{(i_2,i_1i_2\ldots i_k)_3}), \tag{3.22c}$$

$$\lambda_{i_1i_2\ldots i_k} = \max(\lambda_{i_1,i_1i_2\ldots i_k}, \lambda_{i_2,i_1i_2\ldots i_k}). \tag{3.22d}$$

Since (3.14) holds for the new upper-bound signals, either $r_{i_1,i_1i_2\ldots i_k}$ or $r'_{i_1,i_1i_2\ldots i_k}$ can be chosen as the upper-bound signal for $\mathbf{P}_{i_1}$. A similar discussion can be made for the upper-bound signals of plant $\mathbf{P}_{i_2}$. In other words , there exist time instants $\tau_2 > \tau_1$ and $\tau_4 > \tau_3$ such that

$$\tilde{r}(\tau_1) = r_{i_1,i_1i_2\ldots i_k}(\tau_1) + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \Leftrightarrow \tilde{r}(\tau_2) = r'_{i_1,i_1i_2\ldots i_k}(\tau_2) + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \tag{3.23}$$

$$\tilde{r}(\tau_3) = r_{i_2,i_1i_2\ldots i_k}(\tau_3) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon \Leftrightarrow \tilde{r}(\tau_4) = r'_{i_2,i_1i_2\ldots i_k}(\tau_4) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon \tag{3.24}$$

Subtracting (3.21) from (3.20) results in:

$$r'_{i_1,i_1,i_1i_2\ldots i_k} - r'_{i_2,i_1,i_1i_2\ldots i_k} = k_0 e^{\lambda_{i_1i_2\ldots i_k} t} \tag{3.25}$$

where $k_0 > 0$ for $r'_{i_1,i_1,i_1i_2\ldots i_k} > r'_{i_2,i_1,i_1i_2\ldots i_k}$.

$$r'_{i_2,i_1,i_1i_2\ldots i_k} + k_0 > r'_{i_1,i_1,i_1i_2\ldots i_k} \tag{3.26}$$

because $\lambda_{i_1 i_2 \dots i_k} < 0$

It follows from (3.23) that if $\tilde{r}(\tau_2) = r'_{i_2, i_1 i_2 \dots i_k}(\tau_2) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon$

$$\forall \varepsilon > 0 : \exists \varepsilon' = \varepsilon + k_0 + \|\tilde{F}_{i_1} - \tilde{F}_{i_2}\|\tilde{b} > 0 \tag{3.27}$$

and $\exists \tau_5 > \tau_4$ such that

$$\begin{aligned} \tilde{r}(\tau_5) &= r'_{i_2, i_1, i_1 i_2 \dots i_k}(\tau_5) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon' \\ &= r'_{i_2, i_1, i_1 i_2 \dots i_k}(\tau_5) + k_0 + \|\tilde{F}_{i_2}\|\tilde{b} + \|\tilde{F}_{i_1} - \tilde{F}_{i_2}\|\tilde{b} + \varepsilon \end{aligned} \tag{3.28}$$

since $\|\tilde{F}_{i_2}\| + \|\tilde{F}_{i_1} - \tilde{F}_{i_2}\| \geq \|\tilde{F}_{i_1}\|$, then

$$\begin{aligned} \tilde{r}(\tau_5) &\geq r'_{i_2, i_1, i_1 i_2 \dots i_k}(\tau_5) + k_0 + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \\ &\geq r'_{i_2, i_1, i_1 i_2 \dots i_k}(\tau_5) + k_0 e^{\lambda_{i_1 i_2 \dots i_k} \tau_5} + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \\ &= r'_{i_1, i_1, i_1 i_2 \dots i_k}(\tau_5) + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \end{aligned} \tag{3.29}$$

It follows from (3.27) and (3.29) that

$$\begin{aligned} \tilde{r}(\tau_3) &= r'_{i_2, i_1 i_2 \dots i_k}(\tau_3) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon \\ \Rightarrow \exists \tau_6 < \tau_5 : \tilde{r}(\tau_6) &= r'_{i_1, i_1 i_2 \dots i_k}(\tau_6) + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \end{aligned} \tag{3.30}$$

Substituting $\tau_2 = \tau_6$ into (3.23) results in

$$\begin{aligned} \tilde{r}(\tau_6) = & r_{i_2, i_1 i_2 \dots i_k}(\tau_6) + \|\tilde{F}_{i_2}\|\tilde{b} + \varepsilon \Rightarrow \\ \tilde{r}(\tau_1) &= r_{i_1, i_1 i_2 \dots i_k}(\tau_1) + \|\tilde{F}_{i_1}\|\tilde{b} + \varepsilon \end{aligned} \tag{3.31}$$

This implies that if the filtered upper-bound signal meets the smaller boundary signal corresponding to the closed-loop pair $(\mathbf{K}_{i_1 i_2 \dots i_k}, \mathbf{P}_{i_2})$, it will definitely meet the other boundary signal corresponding to the closed-loop pair $(\mathbf{K}_{i_1 i_2 \dots i_k}, \mathbf{P}_{i_1})$, so that $t_i = \min(\tau_1, \tau_3)$. According to Lemma 1, for the higher layer controllers the smallest upper-bound signal associated with each controller and its corresponding plant models is used to be compared to the filtered signal as it results in smaller time instants.

Theorem 2 *Using the switching sequence of Algorithm 2, and the switching instants* $t_s = \min(t_{i-1} + t_d, t_i)$ *where* t_i *represents the time instants given in Lemma 1 (*$t_0 := 0$*) and* t_d *is the maximum time for stability, the system will eventually switch to the correct controller with no more than one unstable switching.*

Proof of Theorem 2: The proof follows from Lemma 1 and the results of Theorem 1 in [4]. It is to be noted that using the above algorithm the system cannot switch to more than one destabilizing controller because the first unstable switching causes the switching route to be ended by a last switching to the matching controller which stabilizes the system.

3.5 Numerical Examples

The following example shows the effectiveness of the proposed multi-layer switching structure in improving the transient response. It is the same example given in Chapter 2 which is used here to compare multi-layer switching to single-layer switching.

Example 6 *Consider the following unstable non-minimum phase plant which was given in Chapter 2:*

$$\boldsymbol{P} = \lambda \frac{s-1}{(s-2)(s+1)}, \quad 1 < \lambda(t) < 6$$

A family of four plant models $\boldsymbol{P}_i = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4\}$ *is then considered as follows*

$$\boldsymbol{P}_1 = \frac{s-1}{(s-2)(s+1)}, \ \boldsymbol{P}_2 = 2\boldsymbol{P}_1, \ \boldsymbol{P}_3 = 4\boldsymbol{P}_1, \ \boldsymbol{P}_4 = 6\boldsymbol{P}_1$$

The high-performance controllers of first layer the same controllers given in Example2:

$$\boldsymbol{K}_1 = \frac{448s^2+450s-18}{31s(s-9)},$$
$$\boldsymbol{K}_2 = \frac{1}{2}\times\frac{448s^2+450s-18}{31s(s-9)},$$
$$\boldsymbol{K}_3 = \frac{1}{4}\times\frac{448s^2+450s-18}{31s(s-9)},$$
$$\boldsymbol{K}_4 = \frac{1}{6}\times\frac{448s^2+450s-18}{31s(s-9)}.$$

The second layer consists of three controllers

$$\boldsymbol{K}_{12} = \frac{45.31s^4+325.5s^3+853.2s^2+1184s+610.3}{s^4-29.81s^3-237.55s^2-515.6s-670.1}$$

$$\boldsymbol{K}_{23} = \frac{22.65s^4+162.7s^3+426.7s^2+591.8s+305.1}{s^4-29.81s^3-237.5s^2-515.6s-670.1}$$

$$\boldsymbol{K}_{34} = \frac{0.2758s^4+53.1s^3+224.35s^2+425s+253.6}{s^4+14.4s^3-124.4s^2-470.9s-872.1}$$

It can be easily verified that controller $\boldsymbol{K}_{12}$ stabilizes the plant models $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and destabilizes $\boldsymbol{P}_3$, $\boldsymbol{P}_4$. Similarly, $(\boldsymbol{K}_{23},\boldsymbol{P}_2)$, $(\boldsymbol{K}_{23},\boldsymbol{P}_3)$, $(\boldsymbol{K}_{34},\boldsymbol{P}_3)$ and $(\boldsymbol{K}_{34},\boldsymbol{P}_4)$ are stable closed-loop pairs while $(\boldsymbol{K}_{23},\boldsymbol{P}_1)$, $(\boldsymbol{K}_{23},\boldsymbol{P}_4)$, $(\boldsymbol{K}_{34},\boldsymbol{P}_1)$ and $(\boldsymbol{K}_{34},\boldsymbol{P}_2)$ are unstable pairs. Assume that initially the actual plant model is $\boldsymbol{P}_1$ and at some point of time it changes to $\boldsymbol{P}_4$. It was seen before that in the single-layer approach the system will switch from $\boldsymbol{K}_1$ to $\boldsymbol{K}_2$, then to $\boldsymbol{K}_3$, and finally to $\boldsymbol{K}_4$. The first two switching instants are unstable. In the multi-layer approach the system will switch from $\boldsymbol{K}_1$ to $\boldsymbol{K}_{23}$, and then to $\boldsymbol{K}_4$. The only unstable switching corresponds to $\boldsymbol{K}_{23}$. Figures 2.3 and 3.3 show that the maximum amplitude of transient response is reduced by 90% compared to the single-layer method of [4].

Now assume that the plant model changes again from $\boldsymbol{P}_4$ to $\boldsymbol{P}_3$. The single-layer method switches from $\boldsymbol{K}_4$ to $\boldsymbol{K}_1$, then to $\boldsymbol{K}_2$, and finally to $\boldsymbol{K}_3$. Two unstable switchings *occur using single-layer method. However, the proposed multi-layer algorithm will switch from $\boldsymbol{K}_4$ to $\boldsymbol{K}_{23}$. The system becomes stable and then switches to $\boldsymbol{K}_3$. It is to be noted that in this case no unstable switching occurs using multi-layer algorithm. Figures 2.4 and 3.4 show that the maximum amplitude of the transient response is 350 times smaller than that of the single-layer method of [4].*

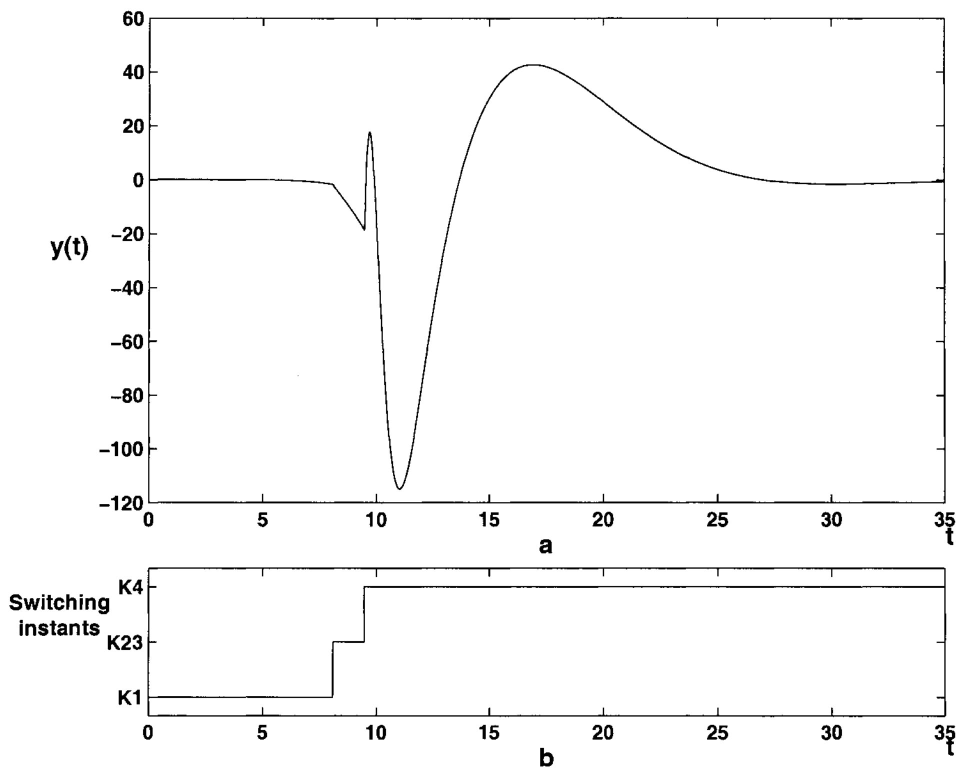

Figure 3.3: Closed-loop simulation results for Example 6, using the multi-layer scheme, when the plant model changes from P_1 to P_4. (a) Output signal; (b) switching instants.

The second example in this chapter compares the results of the multi-layer switching algorithm to a former single-layer counterpart for the practical example given in the

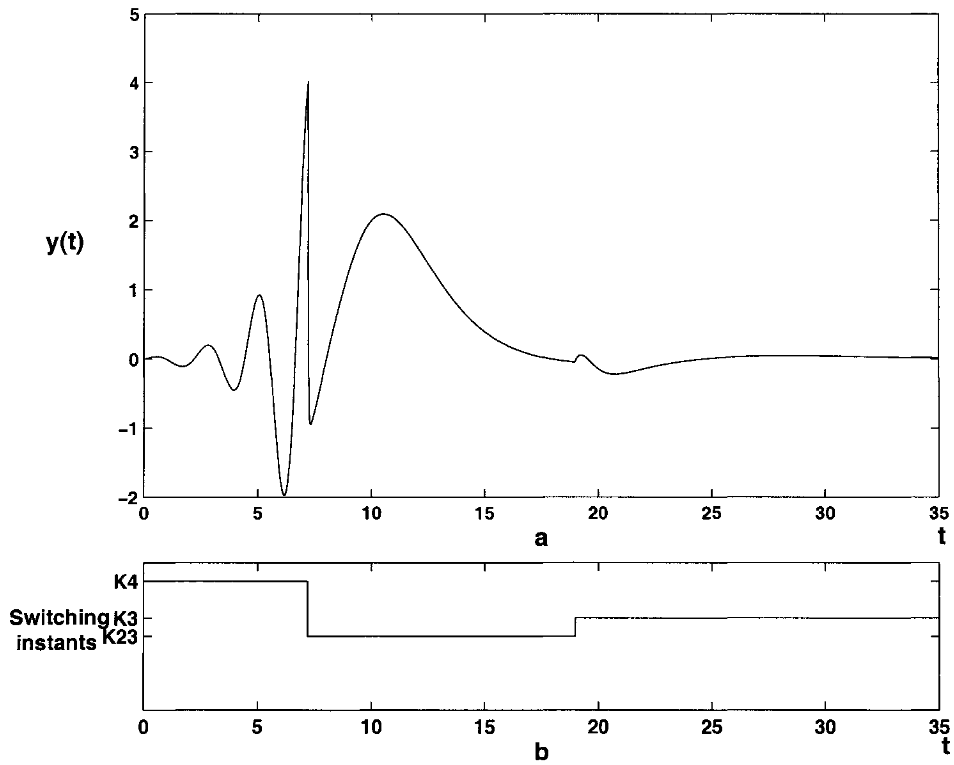

Figure 3.4: Closed-loop simulation results for Example 6, using the multi-layer scheme, when the plant model changes from P_4 to P_3. (a) Output signal; (b) switching instants.

previous chapter.

Example 7 *Assume the model given in Example 3. The first layer is made of the same four controllers given in Example 3. The second layer can be formed by two more controllers which are defined as below:*

Controller 12:

$$S_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad , \quad R_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q_{12} = \begin{bmatrix} -0.1017 & 0.0001648 \\ -0.0002642 & -0.07448 \end{bmatrix} \quad , \quad K_{12} = \begin{bmatrix} -1.679 & 0.007127 \\ -0.009605 & -1.089 \end{bmatrix}$$

Controller 34:

$$S_{34} = S_{12} \quad , \quad R_{34} = R_{12}$$

$$Q_{34} = -Q_{12} \quad , \quad K_{34} = -K_{12}$$

Assume that initially the plant model is $\boldsymbol{P}_1$ and the controller is $\boldsymbol{K}_1$. The plant model suddenly changes to $\boldsymbol{P}_3$ so that the system becomes unstable. This is the same scenario as in Example 3. After the upper-bound is hit, the algorithm switches to $\boldsymbol{k}_{34}$ and stays at this controller until it is known that the closed-loop pair is stable. It then switches to $\boldsymbol{K}_3$. The results can be seen in Figure 3.5 which shows great improvements in the transient response.

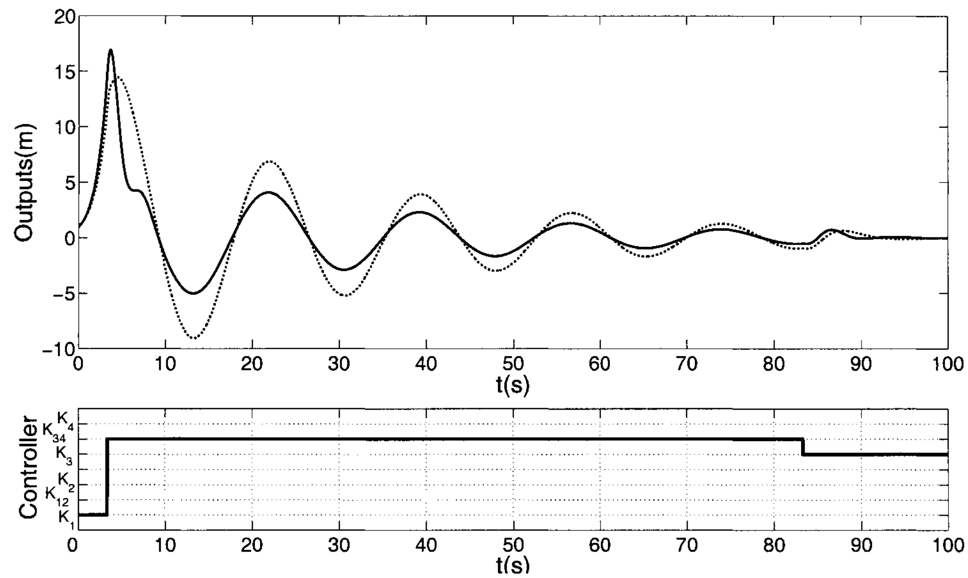

Figure 3.5: Closed-loop simulation results for Example 7, using the multi-layer scheme, when the plant model changes from P_1 to P_3. (a) Output signal; (b) Switching instants.

Chapter 4

Possibility Vectors and Switching Matrices

As discussed earlier, it is assumed that the unknown plant is always a member of a finite set of plants. When the system becomes unstable it will switch to another controller and continues this process until the actual plant is determined. This procedure can be formulated using a so called possibility vector as follows.

4.1 Switching Information

The information gained from the system can be organized using matrices and vectors in order to simplify the switching algorithm and improve it.

Definition 6 *The potential plant models can be represented by a vector:*

$$\Psi(t) = \begin{bmatrix} \psi_1(t) \\ \psi_2(t) \\ \vdots \\ \psi_p(t) \end{bmatrix}, \begin{cases} \psi_i(t) = 0, & \boldsymbol{P}_i \text{ is not possible} \\ \psi_i(t) = 1, & \boldsymbol{P}_i \text{ is possible} \end{cases} \tag{4.1}$$

which is called the possibility vector at time t. It is to be noted that the set of potential plant models at each time is a subset of the family of plants. Also, notice that after each switching instant this set can change, depending on the stability or instability of the system in the switching instant.

Definition 7 *Each controller **K** can be represented by a vector:*

$$\Upsilon_{\boldsymbol{K}} = \begin{bmatrix} \upsilon_1^{\boldsymbol{K}} \\ \upsilon_2^{\boldsymbol{K}} \\ \vdots \\ \upsilon_p^{\boldsymbol{K}} \end{bmatrix}, \begin{cases} \upsilon_i^{\boldsymbol{K}} = 0, & \boldsymbol{K} \text{ destabilizes } \boldsymbol{P}_i \\ \upsilon_i^{\boldsymbol{K}} = 1, & \boldsymbol{K} \text{ stabilizes } \boldsymbol{P}_i \end{cases} \tag{4.2}$$

*which is called the stabilizing representation vector of **K**. Notice that the representation vector for each controller can be obtained off-line.*

Definition 8 *A stable switching to a controller **K** can be denoted by a diagonal matrix:*

$$\Delta_{\boldsymbol{K}} = diag(\Upsilon_{\boldsymbol{K}}) \tag{4.3}$$

*which is defined as the stable switching matrix of **K**.*

Definition 9 *An unstable switching to a controller **K** can be represented by a diagonal matrix:*

$$\nabla_{\boldsymbol{K}} = I - \Delta_{\boldsymbol{K}} \tag{4.4}$$

which is defined as the unstable switching matrix of $\boldsymbol{K}$.

After each switching instant, the number of potential plant models is decreased, regardless of the switching being stable or unstable.

Theorem 3 *If the system switches to controller* $\boldsymbol{K}$ *at time* t_0 *and it is known at time* $t_1 > t_0$ *that the system is unstable and no switching occurs in between* t_0 *and* t_1*, then the possibility vector will be*

$$\Psi(t_1) = \nabla_{\boldsymbol{K}}\Psi(t_0)$$

Proof of Theorem 3: Since no switching occurs for $t_0 \le t < t_1$, the possibilities stay unchanged in this time interval. Consider the representation vector of controller **K** defined by (4.2). At $t = t_1$

$$\begin{aligned} \upsilon_i^{\mathbf{K}} = 1 &\Rightarrow \mathbf{P}_i \text{ is stabilized by } \mathbf{K} \\ &\Rightarrow \mathbf{P}_i \text{ is not possible } \Rightarrow \psi_i(t) = 0, \forall i \in \bar{p} \end{aligned}$$

and

$$\begin{aligned} \upsilon_i^{\mathbf{K}} = 0 &\Rightarrow \mathbf{P}_i \text{ is destabilized by } \mathbf{K} \\ &\Rightarrow \text{ possibility of } \mathbf{P}_i \text{ is not changed} \\ &\Rightarrow \psi_i(t) = \psi_i(t_0), \forall i \in \bar{p} \end{aligned}$$

This implies that $\Psi(t) = \nabla_{\mathbf{K}}\Psi(t_0)$.

Theorem 4 *If the system switches to controller* $\boldsymbol{K}$ *at time* t_0 *and is known at time* $t_1 > t_0$ *that the system is stable and no switching occurs in between* t_0 *and* t_1*, then the possibility vector will be*

$$\Psi(t_1) = \Delta_{\boldsymbol{K}}\Psi(t_0)$$

Proof of Theorem 4: Consider the representation vector of controller $\mathbf{K}$ defined by (4.2). At $t = t_1$

$$\upsilon_i^{\mathbf{K}} = 0 \Rightarrow \mathbf{P}_i \text{ is destabilized by } \mathbf{K}$$

$$\Rightarrow \mathbf{P}_i \text{ is not possible } \Rightarrow \psi_i(t) = 0, \forall i \in \bar{p}$$

and

$$\upsilon_i^{\mathbf{K}} = 1 \Rightarrow \mathbf{P}_i \text{ is stabilized by } \mathbf{K}$$

$$\Rightarrow \text{ possibility of } \mathbf{P}_i \text{ is not changed}$$

$$\Rightarrow \psi_i(t) = \psi_i(t_0), \forall i \in \bar{p}$$

This implies that $\Psi(t) = \Delta_{\mathbf{K}} \Psi(t_0)$.

4.2 Switching Algorithm Using Possibility Matrix

Formulation of the switching algorithm can be simplified by using switching matrices and possibility vectors. The following assumption is made for the development of the algorithm.

Assumption 3

i) For each plant $\boldsymbol{P}_i$ there exists a controller $\boldsymbol{K}_i$ that can stabilize it.

ii) For each plant $\boldsymbol{P}_i$ there exist $p-2$ controllers as $\boldsymbol{K}^1, \boldsymbol{K}^2, \ldots, \boldsymbol{K}^{p-2}$ such that

$$\forall j : 1 \leq j \leq p-2 \Rightarrow$$

$$sum(\nabla_{\boldsymbol{K}^j} \Delta_{\boldsymbol{K}^{j+1}} \Delta_{\boldsymbol{K}^{j+2}} \ldots \Delta_{\boldsymbol{K}^{p-2}} \tilde{\Upsilon}_{\boldsymbol{K}_i}) = 1$$

where $sum(.)$ is the sum of all elements of a vector and $\tilde{\Upsilon}_{\mathbf{K}_i} = \text{not}(\Upsilon_{\mathbf{K}_i})$ and $\text{not}(A)$ is the logical "not" of all elements of A.

Algorithm 3

1) *If the plant was known to be* $\boldsymbol{P}_i$ *and the system became unstable at time t set the possibility vector (4.1) with* $\psi_i(t) = 0$ *and* $\psi_j(t) = 1 : j \neq i$.

2) *If* $sum(\Psi(t)) > 1$ *go to step* 3. *Else find a controller* $\boldsymbol{K}$ *that* $\Upsilon_{\boldsymbol{K}} = \Psi(t)$ *and switch to* $\boldsymbol{K}$ *if it is different from the current controller and stop.*

3) *Switch to a controller* $\boldsymbol{K}$ *such that* $sum(\nabla_{\boldsymbol{K}}\Psi(t)) = 1$.

4) *Wait until the possibility vector changes due to Theorem 3 and 4 then go to step* 2.

Algorithm2 given in the previous chapter is a special case of this algorithm.

Theorem 5 *The controllers needed for algorithm 3 exist if assumption 3 holds.*

Proof of Theorem 5: The switched controller in stage 2 of the algorithm always exists because of part (i) of the assumption. The existence of the controllers in stage 3 can be easily concluded from part (ii) of the assumption.

Theorem 6 *Using the algorithm 3 at most one unstable switching occurs.*

Proof of Theorem 6: Switchings are made only on stage 2 and 3 of the algorithm and the possibility vector is updated on stage 4 due to stability or instability.

If the system switches in stage 2 because $\text{sum}(\Psi(t)) = 1$ only one possibility is true and no instability will happen after that.

If the system switches in stage 3 and becomes unstable the new possibility vector would be $\nabla_{\mathbf{K}}\Psi(t)$ where $\mathbf{K}$ is the switched controller. The algorithm ensures that

$\text{sum}(\nabla_{\mathbf{K}}\Psi(t)) = 1$ which means the algorithm will stop at stage 2 with no more instability as said above.

Example 8 *Consider a family of* 8 *plants with* 6 *layers of controllers in the following sets:*

$$\Phi_1 = \{\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{K}_3, \boldsymbol{K}_4, \boldsymbol{K}_5, \boldsymbol{K}_6, \boldsymbol{K}_7, \boldsymbol{K}_8\}$$

$$\Phi_2 = \{\boldsymbol{K}_{23}, \boldsymbol{K}_{67}\}$$

$$\Phi_3 = \{\boldsymbol{K}_{234}, \boldsymbol{K}_{236}, \boldsymbol{K}_{567}\}$$

$$\Phi_4 = \{\boldsymbol{K}_{1234}, \boldsymbol{K}_{2367}, \boldsymbol{K}_{2567}, \boldsymbol{K}_{5678}\}$$

$$\Phi_5 = \{\boldsymbol{K}_{12345}, \boldsymbol{K}_{23678}, \boldsymbol{K}_{12567}, \boldsymbol{K}_{45678}\}$$

$$\Phi_6 = \{\boldsymbol{K}_{123456}, \boldsymbol{K}_{123458}, \boldsymbol{K}_{123678}, \boldsymbol{K}_{125678}, \boldsymbol{K}_{345678}\}$$

Assume that the plant is stable at $\boldsymbol{K}_4$ *at first. The plant changes to* $\boldsymbol{P}_2$ *at time* t_0 *and the switching algorithm begins. In step* 1 *the possibility vector* $\Psi(t_0) = [1\ 1\ 1\ 0\ 1\ 1\ 1\ 1]'$ *is* *set. Step* 2 *will be ignored because* $sum(\Psi) \neq 1$.

$$\Delta_{\boldsymbol{K}_{125678}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow$$

$$sum(\Delta_{\boldsymbol{K}_{125678}} \Psi(t_0)) = 6 = sum(\Psi(t_0)) - 1$$

According to the above equations, in step 3 *the system will switch to* K_{125678} *and will wait at step* 4 *until it is known to be stable at time* t_1 *and the possibility vector is updated:* $\Psi(t_1) = [1\ 1\ 0\ 0\ 1\ 1\ 1\ 1]'$. *The algorithm goes back to step* 2 *again. The if statement is not met and it moves to step* 3. $sum(\Delta_{\boldsymbol{K}_{12567}} \Psi(t_1)) = 5 = sum(\Psi(t_1)) - 1$ *and the system will switch to* $\boldsymbol{K}_{12567}$. *Assume that at* t_2 *it is known that the closed-loop system has been stable. The new possibility becomes* $\Psi(t_2) = [1\ 1\ 0\ 0\ 1\ 1\ 1\ 0]'$. *Step* 2 *will be ignored again. Let the next controller in Step* 3 *be* $\boldsymbol{K}_{2567}$. *Assume that the system is known to be stable at time* t_3 *with* $\Psi(t_3) = [0\ 1\ 0\ 0\ 1\ 1\ 1\ 0]'$ *in step* 4 *and goes back to step* 2. *Still* $sum(\Psi(t_3)) \neq 1$ *and another step will be taken. The next candidate controller would be* $\boldsymbol{K}_{567}$ *in step* 3. *At time* t_4 *the system is known to be unstable so that* $\Psi(t_4) = \nabla_{\boldsymbol{K}_567} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$. *This time in step* $sum(\Psi(t_4)) = 1$ *which makes the system switch to* $\boldsymbol{K}_2$ *in step* 2 *with* $\Upsilon_{\boldsymbol{K}_2} = \Psi(t_4)$ *and stop.*

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Switching control of a family of plants and its advantages and shortcomings were investigated in this work. Switching control methods can outperform traditional adaptive control techniques when the plant is highly uncertain and/or the *a priori* information required in conventional adaptive control methods is not available. The main shortcoming of switching control methods, however is the bad transient. A brief history of switching control, its development and main problems were given in Chapter 1. The existing switching control methods for a family of plants were referred to as "single-layer switching" as in these methods one controller is assigned to each plant model. Thus, the controllers can be considered as being arranged on a so called "single layer" with a one-to-one correspondence with the plant models.

In *family of plants* approach, a set of plant models represent parameters variations of the plant. It is to be noted that in practice, parameters of the system vary in a compact set. However, one can choose a finite set of values for each parameter to formulate the problem in the family of plants context. Thus, the model of a system with changing dynamics is assumed to be jumping from one member of the family to another. A set of controllers with the same size as the set of plants is then used to stabilize the time varying or uncertain plants. Each controller could stabilize one of the members of the family of plants. Different algorithms are proposed to switch among the controllers using different methods of evaluation based on the error signal or the outputs of the system as a test on performance of each controller until the unknown plant is stabilized. Although it has some advantages compared to the classical adaptive control in the sense that the required *a priori* information about the plant is reduced and it can stabilize plant models with very big changes in their dynamics, the switching control methods suffer from bad transient response. This is discussed in detail in Chapter 2 with illustrative examples to support the claim.

The method proposed in Chapter 3 and reformulated in Chapter 4, targets the bad transient response which is mainly because of switching to destabilizing controllers. The idea behind this method is that by switching to controllers that can stabilize more than one plant model in the family of plants, the number of switchings to destabilizing controllers in the transition to find the correct controller can be reduced. The proposed algorithm uses a special architecture of controllers, namely "multi-layer" structure, so that in each layer a set of properly arranged simultaneous stabilizers is used. The main property of the proposed algorithm is that each switching to a stabilizing controller makes the system one step closer to finding the correct controller and switching to a destabilizing controller can point the decision maker unit to the correct controller. As a result, provided all required simultaneous stabilizers exist in different layers, the system can find the correct controller with at most one unstable switching. If, on the other hand, some of the required simultaneous stabilizers do not exist, one can still take advantage of multi-layer structure and improve the transient response by reducing the number of unstable switchings, as discussed in Chapter 3.

The multi-layer switching algorithm can only be used for families of plants with more than three members because the number of layers in a family of p plants would be $p-2$ and if $p=3$ then the multi-layer structure would be the same as the single-layer one. The simulation results in Chapter 3 show 90% reduction in the magnitude of the transient response compared to the single-layer counterpart proposed by Miller and Davison [4] for a given example, which indicates effectiveness of the multi-layer structure in improving the transient response. It is to be noted that most of the given examples have a family of four plants which is the minimum size for effectiveness of a multi-layer structure as discussed in Chapter 3. The more the number of the plants in the family is the better the results of the algorithm would be compared to the single-layer switching because the average number of unstable switchings would increase in a single-layer structure when the number of plants increases but it is always less than or equal 1 for multi-layer structure.

The main issue in multi-layer switching architecture is designing the higher layer controllers, i.e. the controllers that must stabilize more than one plant model. As shown in Chapter 3 and Chapter 4 there is no need to design all the controllers in higher layers to take advantage of the proposed algorithm. Furthermore, the controllers on higher layers are, in fact, simultaneous stabilizers only, i.e. they are required to stabilize a proper subset of family of plants and do not need to solve the reference signal tracking of disturbance rejection problem. This simplifies the design problem to some extent but in general design of simple simultaneous stabilizers can be a hard task. It is also shown in Chapter 3 that some controllers on higher layers can be used instead of others when a good switching method is proposed. In some cases one or more layers can be omitted completely.

Multi-layer switching structures are a general form of single-layer structures. The work done here can even be used for single-layer methods. There is an assumption in many single-layer algorithms that each controller in the set can stabilize one and only one plant of the family which makes the controllers be sub-optimal instead of optimal controllers for each plant. If the optimal controller of a plant can stabilize another plant as well, the single-layer switching algorithm may be trapped into non-optimal controllers in some cases which is a weak point of them. On the other hand, if the multi-layer switching method is used even for those cases with one layer of controllers, the switching routes will be chosen in a way that the system does not lock onto a controller unless the plant model is clearly known which means that the switching will end up in the optimal controller.

5.2 Future work

The proposed multi-layer switching, like any other switching control method, has different design components, the most important of which are controller design and switching scheme design. Proper design of these components plays an important role in the overall performance of the system.

5.2.1 Design Tools

Digital systems have been widely used in recent years because generally they are cheaper, more reliable, and easy to implement in practice. Discrete-time controllers can be used instead of the continuous-time controllers in multi-layer switching structures using a switching scheme similar to the continuous-time case. It is to be noted that in a sampled-data system, an LTI discrete-time controller with a hold operator act as continuous-time linear time-varying controller for the original continuous-time system and it is known that such controllers have several advantages in decentralized control of large-scale systems. Thus, one can use the discrete-time version of the proposed method in a decentralized control configuration.

Combining the conventional adaptive control methods with multi-layer switching control can also be an interesting area for research. Each controller in the family of plant models can be an adaptive controller that stabilizes a bigger subclass of plants and is more robust compared to a simple controller. Especially, the controllers on higher layers can be designed using the adaptive control theory.

The multi-layer switching control proposed in this thesis uses a set of continuous-time LTI candidate controllers. One can apply the discrete-time version of the switching algorithm for the sampled-data system with a set of discrete-time candidate controllers. Particularly generalized sampled-data hold functions (GSHF) can be used to gain much better control performance [40].

Recently some types of time varying controllers such as GSHFs (Generalized Sampled Data Hold Function) have attracted many researchers. They are functions that can be used instead of a zero order or first order hold function in a discrete time system and are proven to be a way of stabilizing systems. They can sometimes replace a digital controller that has many benefits. Using GSHFs together with multi-layer switching schemes can be very practical in the future of controlling digital systems with uncertainties and changing parameters.

Control of large-scale MIMO systems have resulted in a new area of research called Decentralized Control. Instead of using a complex centralized controller, several smaller controller agents are made to cooperate on stabilization and tracking problem of the whole system. Decentralized controllers have shown more robustness while they are easier to design because they are smaller in degree and number of inputs and outputs. Decentralized switching controllers have been previously used in switching control [33], [13]. They have been proven to be a good way to solve switching control problem for large-scale systems and can be used as powerful tools for designing the controllers in a multi-layer switching structure too. Of course the switching method should be modified for this purpose but the main algorithm remains unchanged.

Unlike single-layer switching control which has a simple structure and algorithm, multi-layer switching structure and algorithm needs more work which can be difficult for cases with a large number of plants in the plant model set. Switching control, and multi-layer switching in particular, has a nature that acts like a state machine or automata. Considering this, the use of DES (Discrete Event Systems) can improve previous single-layer and/or multi-layer switching schemes. It can make the algorithms simpler by using the definitions and rules of DES which are well known in recent years. DES can help improving multi-layer switching in two major ways: minimizing the structure concerning any given specification for the system and programming the supervisor using a computer equipped with DES softwares. Although there can be many controllers on layers of the structure, one may want to use a small subset of them in a way that there exists an algorithm of switching that meets the specifications that should be met for the system, i.e. the maximum number of stable and/or unstable switchings. The algorithms given in this work is in an abstract mathematical form which is easy for a designer to understand but hard to translate to machine languages. When the plant model set has many plants in it, the algorithm can be given to a computer in DES format with a general concept and the computer will generate a supervisor that matches the structure given to it. For example, when there are 10 plant models and a structure of 8 layers, instead of thinking and deciding the switchings at any given instant, DES can design the supervisor using basic rules that are generalized for any number of plants and any given structure which are of course given to the computer using DES.

5.2.2 Other Switching Mechanisms

Although the methods that do not guarantee to switch to each controller at most once like [20] do not seem to be suitable for multi-layer switching, with some changes in the switching scheme they may become useful. They can be combined together with the other methods to make them more robust but the total number of switchings would be increased a lot for sure. It is to be noted that switching to controllers on higher layers gains more information about the system sometimes.

Throughout this work the plants are considered with the same probability of happening but the probability of each plant of the family can be different to others in practice. A switching method that takes this issue into account will show much better results. This is the motivation of assigning probability to each plant. The possibility vectors can be changed to probability vectors. It can be added to single-layer switching also.

Bibliography

[1] A. S. Morse, "Recent problems in parameter adaptive control," *In I. D. Landau, editor, CNRS Colloquium on Development and Utilization of Mathematical Models in Automatic Control*, pp. 733–740, 1983.

[2] B. Martensson, "The order of any stabilizing regulator is sufficient a priori information for adaptive stabilization," *Systems and Control Letters*, vol. 6, no. 2, pp. 87–91, 1985.

[3] D. E. Miller and E. J. Davison, "An adaptive controller which can stabilize any stabilizable and detectable lti system," *In C.I. Byrnes, C. F. Martin and R. E. Saek,(Eds.), Proceedings of the eighth international symposium on mathematics of network and systems,AZ*, 1988.

[4] D. E. Miller and E. J. Davison, "Adaptive control of a family of plants," *In D. Hinrichsen, B. Martensson(Eds.), Control of uncertain systems: Proceedings of an International Workshop, Bremen , West Germany ,Progress in Systems and Control theory*, vol. 6, pp. 197–219, 1989.

[5] D. E. Miller and E. J. Davison, "The self-tuning robust servomechanism problem," *IEEE Transactions on Automatic Control, vol. 34*, vol. 34, no. 5, pp. 511–523, May. 1989.

[6] D. E. Miller and E. J. Davison, "An adaptive tracking problem," *Int. J. Adaptive Control Signal Processing., vol.6*, pp. 45–63, 1992.

[7] D. E. Miller and E. J. Davison, "An adaptive tracking problem with a control input constraint," *Automatica*, vol. 29, pp. 877–887, Jul. 1993.

[8] J. Balakrishnan and K. S. Narendra, "Intelligent control using fixed and adaptive models," *Proceedings of American Control Conference*, pp. 597–601, Jun. 1995.

[9] M. S. Branicky, "Analyzing continuous switching systems: Theory and examples," *Proceedings of American Control Conference*, vol. 3, pp. 3110–3114, Jul. 1994.

[10] M. Chang and E. J. Davison, "Control of unknown systems using switching controllers: An experimental study," *Proceedings of American Control Conference*, vol. 3, pp. 2984–2989, Jul. 1994.

[11] M. Chang and E. J. Davison, "Control of unknown systems using switching controllers: the self-tuning robust servomechanism problem," *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 3, pp. 2833–2838, Dec. 1994.

[12] A. S. Morse, "Supervisory control of families of linear set-point controllers-part 1: Exact matching," *IEEE Transactions on Automatic Control*, vol. 41, no. 10, pp. 1413–1431, Oct. 1996.

[13] A. G. Aghdam and E. J. Davison, "Pseudo-decentralized switching control," *Automatica*, vol. 39, no. 2, pp. 317–324, Feb. 2003.

[14] D. Liberzon, "Switching in systems and control," *Birkhäuser*, 2003.

[15] E. B. Kosmatopoulos and P. A. Ioannou, "A switching adaptive controller for feedback linearizable systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 742–750, Apr. 1999.

[16] L. Rodrigues, A. Hassibi, and J. P. How, "Output feedback controller synthesis for piecewise- affine systems with multiple equilibria," *Proceedings of the American Control Conference*, vol. 3, pp. 1784–1789, Jun. 2000.

[17] L. Rodrigues and J. P. How, "Observer-based control of piecewise-affine systems," *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 2, pp. 1366–1371, Dec. 2001.

[18] L. Rodrigues and J. P. How, "Synthesis of piecewise-affine controllers for stabilization of nonlinear systems," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 3, pp. 2071–2076, Dec. 2003.

[19] K. S. Narendra and J. Balakrishnan, "Improving transient response of adaptive control systems using multiple models and switching," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1861–1866, Sep. 1994.

[20] M. Chang and E. J. Davison, "Adaptive switching control of lti mimo systems using a family of controllers approach," *Automatica*, vol. 35, no. 3, pp. 453–465, Mar. 1999.

[21] M. Chang and E. J. Davison, "Switching control of a family of plants," *Proceedings of American Control Conference*, vol. 1, pp. 1015–1020, Jun. 1995.

[22] M. Chang and E. J. Davison, "Robust adaptive stabilization of unknown mimo systems using switching control," *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 1732–1737, Dec. 1995.

[23] A. S. Morse, "Supervisory control of families of linear set-point controllers," *Proceedings of the 32nd IEEE Conference on Decision and Control*, vol. 2, pp. 1055–1060, Dec. 1993.

[24] A. S. Morse, "Supervisory control of families of linear set-point controllers–part 2: Robustness," *IEEE Transactions on Automatic Control*, vol. 42, no. 11, pp. 1500–1515, Nov. 1997.

[25] D. L. J. P. Hespanha and A. S. Morse, "Hysteresis-based switching algorithms for supervisory control of uncertain systems," *Automatica*, vol. 39, pp. 263–272, Feb. 2003.

[26] D. Borrelli, A. S. Morse, and E. Mosca, "Discrete-time supervisory control of families of two-degrees-of-freedom linear set-point controllers," *IEEE Transactions on Automatic Control*, vol. 44, pp. 178–181, Jan. 1999.

[27] P. Ioannou, "Decentralized adaptive control of interconnected systems," *IEEE Transactions on Automatic Control*, pp. 291–298, Apr. 1986.

[28] D. T. Gavel and D. D. Siljak, "Decentralized adaptive control: structural conditions for stability," *IEEE Transactions on Automatic Control*, pp. 413–426, Apr. 1989.

[29] L. Shi and S. K. Singh, "Decentralized adaptive controller design for large-scale systems with higher order interconnections," *IEEE Transactions on Automatic Control*, pp. 1106–1118, Aug. 1992.

[30] R. Ortega and A. Herrera, "A solution to the decentralized adaptive stabilization problem," *Proceedings of the 31st IEEE Conference on Decision and Control*, vol. 4, pp. 3622–3627, 1992.

[31] W. Changyun, "Decentralized adaptive regulation," *IEEE Transactions on Automatic Control*, pp. 2163–2166, Oct. 1994.

[32] W. Changyun and C. S. Yeng, "Decentralized adaptive control using integrator backstepping," *Automatica*, vol. 33, no. 9, pp. 1719–1724, Sep. 1997.

[33] A. G. Aghdam and E. J. Davison, "Decentralized switching control using a family of controllers approach," *Proceedings of the American Control Conference*, vol. 1, pp. 47–52, 1999.

[34] A. Iftar and U. Ozguner, "An optimal control approach to the decentralized robustservomechanism problem," *IEEE Transactions on Automatic Control*, vol. 34, pp. 1268–1271, Dec. 1989.

[35] G. S. West-Vukovich, E. J. Davison, and P. C. Hughes, "The decentralized control of large flexible space structure," *IEEE Transactions on Automatic Control*, vol. 29, no. 10, pp. 866–879, Oct. 1984.

[36] T. N. Chang and E. J. Davison, "Decentralized control of descriptor systems," *IEEE Transactions on Automatic Control*, vol. 46, pp. 1589–1595, Oct. 2001.

[37] A. S. Morse, D. Q. Mayne, and G. C. Goodwin, "Applications of hysteresis switching in parameter adaptive control," *Proceedings of the 30th Conference on Decision and Control*, vol. 1, pp. 767–772, December 1991.

[38] A. S. Morse, D. Q. Mayne, and G. C. Goodwin, "Applications of hysteresis switching in parameter adaptive control," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1343–1354, Sep. 1992.

[39] D. E. Miller and E. J. Davison, "An adaptive controller which provides an arbitrarily good transient and steady-state response," *IEEE Transactions on Automatic Control*, vol. 36, no. 1, pp. 68–81, Jan. 1991.

[40] P. Kabamba, "Control of linear systems using generalized sampled-data hold functions," *IEEE Transactions on Automatic Control*, vol. 32, pp. 772–783, Sep. 1987.

Chapter 6

Appendix A: Upper Bound

The following results are Lemma 1, Lemma 2 and Theorem 1 of [4].

Lemma 1 Suppose that $u(.) = 0$. Then for every $T > 0$, there exist constants α_{i_1} and α_{i_2}, so that for every initial condition $x(0)$ and every disturbance $w \in PC_\infty$, we have

$$\|x(0)\| \leq \alpha_{i_1} \int_0^T \|y(\tau)\|^2 d\tau + \alpha_{i_2} \sup \|w(t)\|^2.$$

Proof. Fix $T > 0$, and let $x(0)$ and $w \in PC_\infty$ be arbitrary. Since $u(.) = 0$, it follows that

$$y(t) = C_i e^{A_i t} x(0) + [\int_0^t C_i e^{A_i(t-\tau)} E_i w(\tau) d\tau + F_i w(t)] =: y_1(t) + y_2(t).$$

So

$$y_1(t) = y(t) - y_2(t),$$

which means that

$$\beta := \int_0^T \|y_1(\tau)\|^2 d\tau \leq 2 \int_0^T \|y(\tau)\|^2 d\tau + 2 \int_0^T \|y_2(\tau)\|^2 d\tau.$$

Define

$$W_i := \int_0^T e^{A_i' \tau} C' C e^{A_i \tau} d\tau;$$

then

$$\beta = x(0)'W_i x(0).$$

Let α_{i_3} = the smallest singular value of W_i; then

$$\beta - \alpha_{i_3}\|x(0)\|^2 = x(0)'[W_i - \alpha_{i_3} I]x(0) \geq 0,$$

so

$$\beta \geq \alpha_{i_3}\|x(0)\|^2.$$

Hence,

$$\|x(0)\|^2 \leq (2/\alpha_{i_3})\int_0^T \|y(\tau)\|^2 d\tau + (2/\alpha_{i_3})\int_0^T \|y_2(\tau)\|^2 d\tau.$$

Now

$$\|y_2(t)\| \leq (\int_0^t \|C_i e^{A_i(t-\tau)}E_i\| d\tau + \|F_i\|)\sup\|w(\tau)\|,$$

so if we define $\alpha_{i_1} := (2/\alpha_{i_3})$ and

$$\alpha_{i_2} := (2/\alpha_{i_3})\int_0^T [\int_0^t \|C_i e^{A_i(t-\tau)}E_i\| d\tau + \|F_i\|]^2 dt,$$

then the result follows.

Lemma 2. There exist constants γ_{i_1}, γ_{i_2}, γ_{i_3}, and $\lambda_i < 0$, so that for every $\bar{u}$, y_{ref}, $w \in PC$ and initial condition $\tilde{x}(0)$, the solution of (4) satisfies

$$\|\tilde{x}(t)\| \leq \gamma_{i_1}\|\tilde{x}(0)\|e^{\lambda_i t} + \int_0^t e^{\lambda_i(t-\tau)}[\gamma_{i_2}\|\bar{u}(\tau) - \tilde{K}_i(\tilde{y}(\tau) - \tilde{D}_i y_{ref}(\tau))\| + \gamma_{i_3}\|w(\tau)\| d\tau$$

for $t \geq 0$.

Proof. We can write (4a) as

$$\begin{aligned}\dot{\tilde{x}}(t) =& (\tilde{A}_i + \tilde{B}_i\tilde{K}_i\tilde{C}_i)\tilde{x}(t) + \tilde{B}_i\bar{u}(t) + \tilde{E}_i w(t) - \tilde{B}_i\tilde{K}_i(\tilde{y}(t) - \tilde{D}_i y_{ref}(t) - \tilde{F}_i w(t)) \\ =& (\tilde{A}_i + \tilde{B}_i\tilde{K}_i\tilde{C}_i)\tilde{x}(t) + \tilde{B}_i[\bar{u}(t) - \tilde{K}_i(\tilde{y}(t) - \tilde{D}_i y_{ref}(t))] + (\tilde{E}_i + \tilde{B}_i\tilde{K}_i\tilde{F}_i)w(t).\end{aligned}$$

Since $\tilde{A}_i + \tilde{B}_i \tilde{K}_i \tilde{C}_i$ is stable, there exist constants $\gamma_{i_1} > 0$ and $\lambda_i < 0$ so that $\|e^{(\tilde{A}_i + \tilde{B}_i \tilde{K}_i \tilde{C}_i)t}\| \leq \gamma_{i_1} e^{\lambda_i t}$ for $t \geq 0$. Hence,

$$\|\tilde{x}(t)\| \leq \gamma_{i_1} e^{\lambda_i t} \|\tilde{x}(0)\| + \gamma_{i_1} \int_0^t e^{\lambda_i (t-\tau)}.$$

$$[\|\tilde{B}_i\|.\|\tilde{u}(\tau) - \tilde{K}_i(\tilde{y}(\tau) - \tilde{D}_i y_{ref}(\tau))\| + \|\tilde{E}_i + \tilde{B}_i \tilde{K}_i \tilde{F}_i\|.\|w(\tau)\|] d\tau,$$

$$t \geq 0.$$

If we define $\gamma_{i_2} := \gamma_{i_1} \|\tilde{B}_i\|$,and $\gamma_{i_3} := \gamma_{i_1} \|\tilde{E}_i + \tilde{B}_i \tilde{K}_i \tilde{F}_i\|$, then the result holds.

Theorem 1. Suppose that $y_{ref}, w \in PC_\infty$, and that $\|w(t)\| \leq \bar{b}$ for $t \geq 0$. For every initial condition $x(0)$, when Controller 1 is applied to the plant, the closed loop system has the following properties:

(i) the gain eventually remains constant at an element of $\{\tilde{K}_i : i \in \mathbf{p}\}$, and

(ii) the state $\tilde{x}$ is bounded.

Chapter 7

Appendix B: MATLAB Codes

7.1 Multi-layer Switching Code

```
clc clear warning off;

'LOADING THE SYSTEM...';

load system4x2_6.mat;

CTRL{12}=degree_equalizer(CTRL{12},CTRL{1});
CTRL{34}=degree_equalizer(CTRL{34},CTRL{1});

'LOADING COMPLETED.'
' '
[A,B,C,D]=ssdata(PLNT{1});
```

```
[S,R,Q,K]=ssdata(CTRL{1}); [n,m]=size(B); [r,n]=size(C);
s=size(S,1); Dbar=[zeros(r,r);zeros(s,r);eye(r)];

% simulation parameters:
dt=10; N=2000; Yr=[]; T=[]; timeT=10*dt; settling_time=50*dt;

F=0; E=0; epsilon=.1; b=10; %%%%%%%%%%%%%%%% tuning

CTRL_DATA=[1 1 00; 2 2 00; 3 3 00; 4 4 00;
           12 1 2; 34 3 4];
number_of_controllers=size(CTRL_DATA,1);

'CALCULATING THE BOUNDING PARAMETERS...'

lambdaa=inf*ones(34,2); for jj=1:6
    j1=CTRL_DATA(jj,1);
    for j2=1:2
        [jj j2]
        PLNT_DATA=CTRL_DATA(jj,j2+1);
        if PLNT_DATA==00
            alpha1(j1,j2)=inf; alpha3(j1,j2)=inf; gamma1(j1,j2)=inf;
            gamma2(j1,j2)=inf; lambdaa(j1,j2)=inf; Kbar{j1,j2}=inf;
            settling_time(j1,j2)=NaN;
        else
```

```
[Alpha1,Alpha3,Gamma1,Gamma2,Lambdaa,KBar,Settling_time,NormC]=
boundset4x2(PLNT{PLNT_DATA},CTRL{j1},timeT,b,epsilon);
            if size(Alpha1,2)==0
[Alpha1,Alpha3,Gamma1,Gamma2,Lambdaa,KBar,Settling_time,NormC]=
boundset4x2(PLNT{PLNT_DATA},CTRL{j1},timeT,b,epsilon);
            end
alpha1(j1,j2)=Alpha1; alpha3(j1,j2)=Alpha3; gamma1(j1,j2)=Gamma1;
gamma2(j1,j2)=Gamma2; lambdaa(j1,j2)=Lambdaa;
            Kbar{j1,j2}=KBar; settling_time(j1,j2)=Settling_time;
            Norm_C(j1)=NormC;
        end
    end
end

lambda=min(min(lambdaa));

save system4x2_2_BOUND.mat alpha* gamma* lambda* Kbar*
settling_time* Norm_C* 'CALCULATIONS COMPLETED.'

' '
lambdabar=lambda-1;  %%%%%%%%%%%%%%%%%%%%%%%% tuning

'INITIALIZING...'
```

```
 % initial plant
settled=1; state0=state; counter=0;

Y=[]; U1=[]; % for the record



[A,B,C,D]=ssdata(PLNT{1}); number_of_inputs=size(B,2);
number_of_ourputs=size(C,1); [S,R,Q,K]=ssdata(CTRL{1});
x0=zeros(size(A,1),1); z0=zeros(size(S,1),1); y0=zeros(size(C,1),1);
yref=ones(size(C,1),1);
%x0=[0;0]; z0=[0;0;0;0]; y0=0; yref=1;
z=z0; y=y0; xbar0=[x0;z0]; record_rbar=[]; record_r=[];
record_state=[]; condition1=0;

theta=0; state00=state; PP=size(T,2)-1; PP=5000;
PLANT=PLNT{new_state}; 'INITIALIZING COMPLETED.' ' '

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% END OF INITIALIZATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULATION BEGINS

'SIMULATIONS STARTED...'
for i=1:PP %size(T,2)-1

    [state i-PP]
```

```
[A,B,C,D]=ssdata(PLNT{1}); a=size(A,1); u_dim=size(B,2);
[S,R,Q,K]=ssdata(CTRL{state}); s=size(S,1);

if state0~=state
    settled=0; % we are in switching mode
    counter=0; % ready to count
else
    counter=counter+1; % counting just started
end

if counter*dt>max(settling_time(state,:)*2); %%%%%%% TIME
    settled=1;
end

state0=state; % history of state

x0=xbar0(1:a,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONTROLLER PHASE I
if i*dt<=timeT % time T in the paper
    u=zeros(number_of_inputs,1); u0=u;
    y0=y;
    [x,y]=timeresponse(PLANT,x0,u0,u,dt);
```

```
    z0=z; % phase one, controller is off and u=0 goes to plant
    z=z*0;
    xbar=[x;z];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONTROLLER PHASE II
else
    [z_void,u]=timeresponse(CTRL{state},z0,y0-1,y-1,dt);
    y0=y; xbar0=xbar; z0=z; % previous amount of the parameters
    SYS=feedbackseries(PLANT,CTRL{state});
    [xbar,y]=timeresponse(SYS,xbar0,yref,yref,dt);
end

x=xbar(1:a,:);
z=xbar(a+1:a+s,:);
zdot=(z-z0)/dt;
ubar=[u;zdot]; ybar=[y;z;yref];
Y=[Y y];
U1=[U1 u];

if i*dt<=timeT % time T in the paper
        theta=theta+norm(y,2)^2*dt; %page 203, integrating...
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filtered Signal
if i*dt<=timeT % Time T in the paper
    rbar=0;
else
    rbar=filtering(rbar,yref,ybar,Dbar,lambda,lambdabar,dt);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% END of Filtered Signal




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Auxiliary Signals
for jj=1:number_of_controllers
    j1=CTRL_DATA(jj,1);
    for j2=1:2
        if alpha1(j1,j2)<inf

            if i==1
                r(j1,j2)=0;
            elseif i*dt<=timeT % Time T in the paper
r(j1,j2)=auxiliary(r(j1,j2),yref,ubar*0,ybar,Kbar{j1,j2},
Dbar,lambdaa(j1,j2),gamma2(j1,j2),dt);
r0(j1,j2)=r(j1,j2);
                rj1j2=r(j1,j2);
```

```
elseif (i-1)*dt<=timeT % means (i+1)*dt>=timeT and i*dt<=timeT
                    mu(j1,j2)=sqrt(alpha1(j1,j2)*theta);
r(j1,j2)=gamma1(j1,j2)*exp(lambdaa(j1,j2)*timeT)*mu(j1,j2)+r(j1,j2);
                else
r(j1,j2)=auxiliary(r(j1,j2),yref,ubar,ybar,Kbar{j1,j2},
Dbar,lambdaa(j1,j2),gamma2(j1,j2),dt);
                end

            else % when alpha1==inf ==> no such combination
                r(j1,j2)=NaN;
            end

        end%j2=1:4
    end%j1=1:24
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End of Auxiliary Signals




    record_rbar=[record_rbar;rbar];
    record_r=[record_r;min(r,[],2)'];
    record_state=[record_state;state];
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Conditions and Switching
    condition=rbar>(min(r(state,:)*Norm_C(state)+ 20 ));
     states4x2;  % this is for multi-layer simulations
%      if condition  % this is for single-layer simulations
%          state=state+1;
%          if state>4
%              state=1;
%          end
%      end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End of Conditions and Switching

end 'SIMULATIONS COMPLETED.' ' '

save iRESULT
```

7.2 Bounding Function Parameters Code

```
function [alpha1,alpha3,gamma1,gamma2,lambda,Kbar,
settling_time,NormC]=boundset4x2(PLNT,CTRL,T,b,epsilon)
[A,B,C,D]=ssdata(PLNT);
%Adaptive Control of a Family of Plants,
%Miller and Davison, Page 201-202
```

```
[S,R,Q,K]=ssdata(CTRL); [n,m]=size(B); [r,n]=size(C); s=size(S,1);

Abar=[A zeros(n,s);zeros(s,n) zeros(s,s)]; Bbar=[B
zeros(n,s);zeros(s,m) eye(s)]; Cbar=[C zeros(r,s);zeros(s,n)
eye(s);zeros(r,n) zeros(r,s)]; Dbar=[zeros(r,r);zeros(s,r);eye(r)];

Kbar=[K Q -K;R S -R];
%Kbar=[-K Q K;-R S R];
%syms x;
%W=eval(int(expm(A*x)*expm(A*x),0,T));

Chat=[-(2*A)' eye(n) zeros(n,n) zeros(n,n);
      zeros(n,n) -(2*A)' zeros(n,n) zeros(n,n);
      zeros(n,n) zeros(n,n) 2*A eye(n);
      zeros(n,n) zeros(n,n) zeros(n,n) zeros(n,n)];
eChat=expm(Chat*T);

W=(eChat(2*n+1:3*n,3*n+1:4*n));

alpha3=svds(W,1,0); %smallest singular value of W
alpha1=2/alpha3;
alpha2=0; % for the case that F==E==0
```

```
if size(S,2)==0
    S=zeros(1,1); R=eye(1); Q=zeros(1,1);
    Kbar=[K Q -K;R S -R];
end

lambda=max(real(eig(Abar+Bbar*Kbar*Cbar)));

gamma1=0;
for i1=1:2 dt=.01;

for i=1:2000
    t=i*dt;
    matrix_plot(i)=norm(expm((Abar+Bbar*Kbar*Cbar)*t));
    gamma_plot(i)=gamma1*exp(lambda*t);
end

[max_matrix0,i_max]=max(matrix_plot-gamma_plot);
max_matrix=max(matrix_plot); t_max=dt*i_max;

gamma=max_matrix/(exp(lambda*t_max)); gamma1=gamma;
end
```

```
gamma2=gamma1*norm(Bbar,2); gamma3=0;

SYS=feedbackseries(PLNT,CTRL); settling_time=settling(SYS);

NormC=norm(Cbar,2);
```