Otto-von-Guericke-Universität Magdeburg

Diplomarbeit

A first step towards Adaptive Identification and Control of time-varying plants

von
Jonathan Epperlein

5. Juni 2008

Eingereicht an: Otto-von-Guericke-Universität Magdeburg
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungstechnik
Universitätsplatz 2
Postfach 4120, 39016 Magdeburg
Deutschland

Prüfer: Prof. Dr. rer. nat. Dietrich Flockerzi
Fachgruppe für System- und Regelungstheorie
Max-Planck-Institut Magdeburg

Zweitprüfer: Prof. Dr.-Ing. Jörg Raisch
Fachgebiet für Regelungssysteme
Fakultät für Elektrotechnik und Informatik
Technische Universität Berlin

Betreuer: Prof. Kumpati S. Narendra
Center for Systems Science
Yale University, New Haven, CT

Declaration of originality

I hereby declare that this thesis and the work reported herein was composed and originated entirely by myself. Information derived from the published and unpublished work of others is acknowledged in the text and a list of references is given in the bibliography.

Acknowledgements

...per aspera ad astra.

Expressing gratitude is one of my favourite things to do, so I will not miss this opportunity to do so.

First of all, I want to thank Prof. Narendra, who kindly agreed to take me on as a student for such a short period of time, who made it possible for me to come to a great place like Yale and who taught me innumerable priceless lessons in his very own way.
Zhiling Tian was a pleasant and patient office mate for me.
I also am deeply grateful to Prof. Robert Shorten of the Hamilton Institute, Maynooth, Ireland, who is and was supporting me in many ways and especially made the contact to Prof. Narendra.

Work is one thing, and the time you do not work is another. Nevertheless, if your social environment does not provide for distraction, recreation and fun, work suffers. And you suffer even more. So "thanks a million" to Sergio Hoyer, Mark Hauptmann, Frank Limbrock, Theresa Reinold, Ruth Ditlmann and Liz Evans for being great friends, to Kristina Mõis for basically the same, but from the view of a real Yale undergrad, to Lindsey Noe and her awesome parents for making the incredible Southern hospitality credible, Maggie Stahl for even more hospitality and that initial spark, Raj Dhillon for a lot of new ideas how I could spend my time and of course to Emily Miller who made an additional sun shine for me ever since I met her.

Da mit dieser Arbeit auch mein Studium endet, will ich auch auf deutscher Seite ein paar Dankeschöns loswerden.
Ich hatte großartige Kommilitonen in Ansgar Bohmann, Christian Borchert, PeterBorngräber, Nickel Dittrich, Johannes Dold, André Franz, Florian Knorn und Tina Paschedag, in Prof. Jörg Raisch und Prof. Dietrich Flockerzi zwei wunderbare Lehrer, die mich weit über das Notwendige hinaus unterstützt haben und in Janine Holzmann schlichtweg die beste Sekretärin, die man sich nur wünschen kann. Danke dafür!

Zuletzt noch Dank an die, die den eigentlich größten Anteil an allem was ich so tue haben, die mir sicherlich am meisten gefehlt haben und denen ich diese Arbeit dann auch irgendwie widmen will: Meiner Familie Amélie, Andreas, Lilith, Jascha und Ruben!

Jonathan Epperlein, New Haven
May 28, 2008

Abstract

The need to account for time-variations in plant behaviour has become the main reason to resort to Adaptive Control and Identification schemes. While (Multiple-Model) Adaptive Control schemes have proven efficient in time-varying environments, their applicability relies on the heuristic assumption that the variations are still slow enough to allow convergence of some kind. In this thesis, a method is presented which, for the first time, explicitly incorporates time-variations on time-intervals that are too small to allow convergence. In particular, a discrete-time plant which switches arbitrarily between a finite set of ν linear subsystems can be identified and, with the minor restriction that a subsystem stays active for at least as many time-steps as the input delay d, controlled. This is achieved by application of an adaptive algorithm that causes ν models to converge to the subsystems asymptotically.

Zusammenfassung

Die Notwendigkeit, Zeitvariationen im Verhalten der Strecke zu berücksichtigen, ist zum Hauptgrund für die Verwendung adaptiver Regelung geworden. Obwohl adaptive Regelungsmethoden (u.U. unter Verwendung mehrerer Streckenmodelle) sich in zeitvarianten Umgebungen als sehr wirksam gezeigt haben, beruht ihr Einsatz immer noch auf der heuristischen Annahme, dass die Variationen so langsam sind, dass die Adaption zumindest "fast" konvergieren kann. In dieser Arbeit wird eine Methode vorgestellt, die erstmalig Parametervariationen, die zu schnell sind als dass ein adaptives Schema konvergieren könnte, explizit einbezieht. Konkret wird gezeigt, wie eine zeitdiskrete Strecke, die beliebig zwischen ν linearen Subsystemen hin- und herschaltet, identifiziert und (mit der kleinen Einschränkung, dass jedes Subsystem mindestens für eine Zeitspanne, die der Eingangsverzögerung d entspricht, aktiv bleibt) geregelt werden kann. Dies wird durch die Anwendung eines adaptiven Algorithmus ermöglicht, der ν Streckenmodelle asymptotisch zu den Subsystemen konvergieren läßt.

Table of contents

1 Introduction

Time-variations and Adaptive Control

It is intrinsic to all practical applications of control theory that the parameters of the systems one is dealing with are not known exactly. If the possible errors in the parameter values are deemed negligible, no changes in control system design are necessary, but if the uncertainties are large enough to possibly cause performance degradation, measures have to be taken.

Two different main approaches can be followed. The subfield of Robust Control can be roughly described by that: A fixed controller is designed such that it guarantees stability or even a certain performance for all plants with parameter values in the range of the uncertainties.
In the field of Adaptive Control, on the other side, the control system is enabled to monitor its performance and change its parameters in a way that ensures that stability and satisfactory performance are achieved. (Excellent and comprehensive historical overviews as well as discussions of the uses of Adaptive Control can be found in the classic literature, e.g. in [GS84] for the discrete-time and in [NA89] for the continuous-time case.)

From its onset in the early 1960s on, it was also realized that Adaptive Control provides a very good way to deal with slow but persistent time variations like thermal drift in resistors or the wear and tear of mechanical elements. While these effects can cause the parameter to vary over a very large range and by that negate the use of Robust Control, an Adaptive Control scheme can track the parameter values, provided that the adaptation takes place on a considerably faster time-scale than the drift.

Because inherently a nonlinear control scheme, theoretical results about Adaptive Control only established asymptotic properties, typically convergence of output and/or parameter errors to zero. Performance assessment of adaptive controllers in the transient phase, on the other hand, relied heavily on simulations and experiments. It nevertheless is still commonly accepted that Adaptive Control is extremely powerful if the initial parameter estimates are good and the time-variations, if present at all, are slow.

With the development of Multiple-Model schemes ([MGHM92, NBC95, NB97, NX00]) in the 1990s, the initial value problem was somewhat attenuated by distributing a number of models over the range of possible parameter vectors. Since convergence was sped up by those schemes too, they were also regarded as more apt to solve time-varying

problems than the classical single-model schemes. Incorporation of parameter variation with time, however, still was merely implicit by assuming the adaptation would be "fast enough," and strict theoretical results were only obtained for time-invariant cases.

A closer look at the application of adaptive techniques to problems with time-variations reveals that actually, after every change in plant dynamics a completely new adaptation problem is solved. If adaptation has not converged but still is in a transient phase when the next change occurs, the response of the overall system jumps from transient to transient and no convergence takes place.

Since environmental influences like extreme gusts in aircraft control, component failure or loose contacts in electrical networks, rupture of mechanical elements and so on can cause time-variations that are too fast for any adaptive scheme to converge, the demand for a framework that explicitly incorporates time-variance is high.

The algorithm presented in [FN06] can be seen as a first step toward the solution of this problem. The time-variant plant is thereby abstracted as a finite number ν of linear subsystems between which the plant switches arbitrarily. It is a complete novelty with regard to the available techniques because it does not recast the problem of parameter changes as individual problems to be solved sequentially but as ν problems that are solved *simultaneously.* The time-variance is incorporated explicitly in the plant model and the algorithm thus no longer depends on the intervals of constant parameters to be long enough for convergence.
In particular, the algorithm enables us to set up ν models for the ν subsystems and see one model converge to each subsystem, where no more information is assumed to be accessible than in the classical case, i.e. the structure of the plant models and measurements of input and output.

Outline

The development is traced from the classical single-model Adaptive Control and Identification schemes of the 1960s-1980s over the Multiple-Model schemes of the 1990s to the simultaneous adaptive identification algorithm.

After some mathematical preliminaries and the representations of discrete-time dynamical systems to be used in the sequel are presented in **Chapter 2**, the first basic results of Adaptive Identification and Control are derived in **Chapter 3** and illustrated with some simulations. **Chapter 4** deals with the Multiple-Model schemes of the 1990s. The basic idea is presented, stability is proven and more simulations show the superiority to single-model algorithms in the presence of time-variations.

Finally, **Chapter 5** presents the new method. The control algorithm is fairly easy and is presented in Section 5.3. However, the control algorithm relies crucially on the existence of an identification algorithm that causes ν adaptive models to converge to the

ν subsystems asymptotically. One such algorithm is presented in Section 5.4. Sadly, no formal proofs of stability could be given for the resulting time-variant, highly-coupled nonlinear system. Instead, empiric findings are supported with simulations. The chapter is rounded off with simulations of the combination with the control algorithm.

Chapter 6 concludes the thesis by summarizing the findings and giving a short outlook to possible further developments.

2 Mathematical Preliminaries

2.1 Definitions

First of all, we need to clarify a few notations and terms. Since most of it is common control theory and mathematical terminology, this section is kept rather short.

As usual, we denote the set of the real numbers by $\mathbb{R}$, the nonnegative real numbers by $\mathbb{R}^+$, the set of integers and positive integers (or natural numbers) by $\mathbb{Z}$ and $\mathbb{Z}^+$ respectively, the set of column vectors with n real entries by $\mathbb{R}^n$ and the set of matrices with real entries and with n rows and m columns by $\mathbb{R}^{n\times m}$. Scalars and column vectors will be denoted by lower case letters like x, row vectors will be written as the transpose of a column vector: x^T, and matrices will be denoted by capital letters. For a quadratic matrix $P \in \mathbb{R}^{n\times n}$, $P > 0$ ($P \geq 0$) means that P is positive (semi-)definite and that the matrix is Schur means that all eigenvalues lie inside the open unit disc of the complex plane. For two matrices $P, Q \in \mathbb{R}^{n\times n}$, $P > Q$ ($P \geq Q$) means that the difference $P-Q$ is positive (semi-)definite. Finally, I denotes the identity matrix in appropriate dimension.

Control Theory and also the subset of Adaptive Control Theory deal with dynamical systems. The concept of a dynamical system as a system with memory is very general, but our attention will be restricted to discrete-time systems with one input and one output whose dynamics are governed by ordinary difference equations as defined below.

Definition 2.1 (Single-input single-output dynamical systems)**.** A discrete-time dynamical system is represented by a set of n difference equations and an output function g

$$x(k+1) = f(x(k), k) \qquad x(k_0) = x_0 \tag{2.1}$$

$$y(k) = g(x(k), k) \tag{2.2}$$

or a n-th order difference equation

$$y(k+n) = \tilde{f}(y(k+n-1), \ldots, y(k), k) \qquad y(k_0) = y_0, \ldots, y(k_0+n-1) = y_{n-1}. \tag{2.3}$$

$x, x_0 \in \mathbb{R}^n$ are called the *state* and *initial state*, respectively, $f : \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}^n$ and $\tilde{f} : \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}$ are referred to as the *right hand sides* and $g : \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}$ is the *output function*.
If the functions f, g or $\tilde{f}$ do not depend on the time k, we call the system *autonomous*.
If the functions f, g or $\tilde{f}$ are linear, we call the system *linear*.

If the dependency of f, g or $\tilde{f}$ on the time k is implicit through a known external signal u, i.e. if

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \qquad x(k_0) = x_0 \\ y(k) &= g(x(k), u(k)k) \end{aligned}$$

or

$$\begin{aligned} y(k+n) = \tilde{f}(y(k+n-1), \ldots, y(k), u(k+m), \ldots, u(k)) \\ y(k_0) = y_0, \ldots, y(k_0+n-1) = y_{n-1}, \end{aligned}$$

$m \leq n$, then the system is non-autonomous but *time-invariant* and we call $u : \mathbb{Z}^+ \to \mathbb{R}$ the *input* of the system.
Solutions are denoted as $x(k; x_0, k_0)$ $(y(k; y_0, \ldots, y_{n-1}, k_0))$ or, if initial time and state are evident from the context or do not matter, short as $x(k)$ $(y(k))$.
A state x_e for which $x_e = f(x_e, k)$ $\forall k \in \mathbb{Z}^+$ is called *equilibrium state* of the dynamical system (2.1).

Remark 2.2. For the sake of simplicity and without loss of generality, we can assume that if equilibria exist, one of them is the origin. Otherwise we can simply shift the state-space: If $x_e \neq 0$ is an equilibrium, then define $\xi := x - x_e$. In the new state ξ, $\xi = 0$ is an equilibrium.

A system which is linear and time-invariant is called a *LTI system* and we will present different ways of representing such a system in Section 2.2.

When we are talking about signals, the spaces ℓ^2 and ℓ^∞ will appear. They are defined generally in the context of ℓ^p spaces of sequences of real numbers:

Definition 2.3 (ℓ^p spaces)**.** For a fixed $p \in [1, \infty)$, we define the ℓ^p norm of a sequence $f : \mathbb{Z}^+ \to \mathbb{R}$ as

$$\|f\|_p := \left(\sum_{k=1}^{\infty} |f(k)|^p \right)^{1/p} < \infty \tag{2.4}$$

if the limit exists and say that $f \in \ell^p$.
For the case $p = \infty$ we have $f \in \ell^\infty$ if

$$\|f\|_\infty := \sup_{k \in \mathbb{Z}^+} |f(i)| \tag{2.5}$$

exists, that is if the sequence f is bounded.

The following theorem summarizes some rather obvious properties of the ℓ^p spaces which will be used in the proofs throughout this thesis without further reference to this theorem:

Theorem 2.4. *(i) If $f \in \ell^p$, $p \in [1, \infty)$, then $\lim_{k \to \infty} f(k) = 0$,*

(ii) $\ell^p \subset \ell^\infty \ \forall p \in [1, \infty)$,

(iii) $\ell^p \subset \ell^q$ *if* $p, q \in [1, \infty)$ *and* $p < q$, *and*

(iv) If $f \in \ell^p$, $p \in [1, \infty)$ *and* $g \in \ell^\infty$, *then* $fg \in \ell^p$.

Proof. (i) Assume that $f(k) \not\to 0$. Then, for some $\varepsilon > 0$, there exists an infinite number of indices k_i, $i \in \mathbb{Z}^+$, such that $|f(k_i)| > \varepsilon$. It then follows, that

$$\|f\|_p = \lim_{N\to\infty} \left(\sum_{k=1, k\neq k_i}^{N} |f(k)|^p + \sum_{i=1}^{k_i \leq N} |f(k_i)|^p \right)^{1/p} \geq \lim_{N\to\infty} (N'\varepsilon) = \infty,$$

where $N' = \max\{i \,|\, k_i \leq N\}$. The inequality follows, because the first sum is nonnegative and $N' \to \infty$ as $N \to \infty$ follows, because there is an infinite number of k_i.

(ii) That $\ell^\infty \neq \ell^p$ is clear from the counterexample $f(k) = k^{-\frac{1}{p}}$, which is clearly bounded by 1 but whose ℓ^p-norm is the p-th root of the harmonic series, which, as is known, diverges. The statement thus is proven by showing that from $f \notin \ell^\infty \Rightarrow f \notin \ell^p$. Since f is not bounded, there exists a sequence $Z' = \{z_1, z_2, \ldots\} \subset \mathbb{Z}^+$ with $z_i < z_{i+1}\ \forall i$ and $\lim_{i\to\infty} z_i = \infty$ such that $\lim_{i\to\infty} |f(z_i)| = \infty$. For the ℓ^p-norm of f it follows, that

$$\begin{aligned}
\left(\sum_{k=1}^{\infty} |f(k)|^p \right)^{1/p} &= \left(\sum_{k\in\mathbb{Z}^+\setminus Z'} |f(k)|^p + \sum_{k\in Z'} |f(k)|^p \right)^{1/p} \\
&\geq \left(\sum_{k\in Z'} |f(k)|^p \right)^{1/p} \\
&\geq \left(\lim_{i\to\infty} |f(z_i)|^p \right)^{1/p} \\
&= \infty.
\end{aligned}$$

The first inequality thereby follows, because the first sum is nonnegative, the second one follows, because only nonnegative terms in the sum are discarded.

(iii) This can be seen as a generalization of item (ii) and is proven as follows: Let $f \in \ell^p$. Then it follows from (ii), that $f \in \ell^\infty$, so let $K > 0$ be an upper bound on f. Then we have

$$\begin{aligned}
\left(\sum_{k=1}^{\infty} |f(k)|^q \right)^{1/q} &= \left[\left(\sum_{k=1}^{\infty} |f(k)|^p |f(k)|^{q-p} \right)^{1/p} \right]^{p/q} \\
&\leq K^{p\frac{q-p}{q}} \left[\left(\sum_{k=1}^{\infty} |f(k)|^p \right)^{1/p} \right]^{p/q} = K' \left(\|f\|_p \right)^{p/q} < \infty.
\end{aligned}$$

The first inequality thereby follows, because $(q-p) > 0$.

(iv) Let $K > 0$ be an upper bound of g. Then

$$\left(\sum_{k=1}^{\infty} |f(k)g(k)|^p\right)^{1/p} \leq K \left(\sum_{k=1}^{\infty} |f(k)|^p\right)^{1/p} < \infty.$$

□

The main interest in the design of controllers or identifiers is, of course, the stability of the resulting overall system. While there are many different notions of stability, the only ones that will be used here are the following of asymptotic and exponential stability.

Definition 2.5 (Asymptotic and exponential stability)**.** An equilibrium x_e of system (2.1) is called *asymptotically stable*, if there exists a neighborhood of x_e such that every solution starting in that neighborhood tends to x_e if k tends to infinity without leaving another, possibly bigger, neighborhood of x_e.
If that holds for the whole state space $\mathbb{R}^n$, then x_e is called *globally* asymptotically stable.
If the solutions tend to x_e with an at least exponential rate, the equilibrium is called *(globally) exponentially stable*.
If a dynamical system possesses an asymptotically (exponentially) stable equilibrium, we simply call the system asymptotically (exponentially) stable.

This rather informal definition is sufficient for the following treatments, for formal definitions see any book on mathematical control theory, e.g. Section 2.4.1 in [NA89]. A very simple concept of stability is the so-called bounded-input bounded-output (BIBO) stability which is easily defined with help of the space ℓ^∞:

Definition 2.6 (BIBO stability)**.** A system is called *bounded-input bounded-output (BIBO) stable* if a bounded input u leads to a bounded output y: $u \in \ell^\infty \Rightarrow y \in \ell^\infty$.

2.2 System models for linear systems

In the history of adaptive identification and control it soon turned out that the way the plant and the models are parametrized is a key ingredient to the successful design of stable algorithms. In this section, we develop the model we will be using throughout this thesis, the so-called *static parametric model (SPM)*, from the state-space representation and the difference equation representation of a liner time-invariant discrete-time plant model.

2.2.1 State-space and difference equation representation

Since the state-space representation contains the most information about the system and its dynamics it is our starting point for the further derivations. Let the quadruple $[A, b, c^T, g]$ be an observable realization (i.e. (A, c^T) is an observable pair, cf. Theorem A.2), so the system is given by

$$x(k+1) = Ax(k) + bu(k) \tag{2.6a}$$
$$y(k) = c^T x(k) + gu(k). \tag{2.6b}$$

The term g is thereby referred to as the direct throughput term, because it describes the immediate effect of the input on the output. Since in most applications if u is used to control the system, the output is measured first and the input is computed using the measurement, the input cannot have an immediate effect and g will usually be equal to zero.

We can propagate the output as

$$\begin{aligned}
y(k+1) &= c^T x(k+1) = c^T \left[Ax(k) + bu(k)\right] + gu(k+1) \\
y(k+2) &= c^T x(k+2) = c^T \left[A^2 x(k) + Abu(k) + bu(k+1)\right] + gu(k+2) \\
&\vdots \\
y(k+n-1) &= c^T x(k+n-1) = \\
&\quad c^T \left[A^{n-1} x(k) + A^{n-2} bu(k) + \ldots + bu(k+n-2)\right] + gu(k+n-1).
\end{aligned} \tag{2.7}$$

This can be written in the compact form

$$\mathrm{y}_{n-1} = \mathcal{O} x(k) + \bar{B} \mathrm{u}_{n-1}, \tag{2.8}$$

where

$$\begin{aligned}
\mathrm{y}_{n-1} &= \begin{bmatrix} y(k) & y(k+1) & \ldots & y(k+n-1) \end{bmatrix}^T \\
\mathrm{u}_{n-1} &= \begin{bmatrix} u(k) & u(k+1) & \ldots & u(k+n-1) \end{bmatrix}^T \\
\mathcal{O} &= \begin{bmatrix} c^T \\ c^T A \\ \vdots \\ c^T A^{n-1} \end{bmatrix} \\
\bar{B} &= \begin{bmatrix} g & 0 & 0 & \ldots & 0 \\ c^T b & g & 0 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ c^T A^{n-2} b & \ldots & \ldots & c^T b & g \end{bmatrix}.
\end{aligned}$$

$\mathcal{O}$ is the *observability matrix* (cf. Theorem A.2) of system (2.6) and is invertible since we assumed the system to be observable. From Equation (2.8) we obtain

$$x(k) = \mathcal{O}^{-1}(\mathrm{y}_{n-1} - \bar{B}\mathrm{u}_{n-1}) \tag{2.9}$$

and with that

$$\begin{aligned} y(k+n) &= c^T \left[A^n x(k) + A^{n-1}bu(k) + \ldots + bu(k+n-1)\right] + gu(k+n) \\ &= c^T A^n \mathcal{O}^{-1}(\mathrm{y}_{n-1} - \bar{B}\mathrm{u}_{n-1}) + \begin{bmatrix} c^T A^{n-1}b & \ldots & c^T b & g \end{bmatrix} \mathrm{u}_n \\ &= \sum_{i=0}^{n-1} \lambda_i y(k+i) + \sum_{i=0}^{n} (\bar{\beta}_i + \bar{\gamma}_i) u(k+i), \end{aligned} \tag{2.10}$$

where λ_i are the entries of the row vector $c^T A^n \mathcal{O}^{-1}$ (actually they are the negative coefficients of the characteristic polynomial of A, too. This can be seen expanding A^n using the Cayley-Hamilton Theorem, but has no further importance), $\bar{\beta}_n = 0$ and the other $\bar{\beta}_i$ are the entries of the row vector $-c^T A^n \mathcal{O}^{-1} \bar{B}$, $\bar{\gamma}_i = c^T A^{n-1-i} b$ for $i = 0, \ldots, n-1$ and $\bar{\gamma}_n = g$.
Now we define d as the greatest integer such that $c^T b = c^T Ab = \ldots = c^T A^{d-1} b = 0$, or, more formally

$$d = \begin{cases} 0 & \text{if } g \neq 0 \\ \max\{z \in \mathbb{Z}^+ | c^T A^{\zeta-1} b = 0 \quad \forall \zeta = 1, \ldots, z\} & \text{if } g = 0 \end{cases} \tag{2.11}$$

Then it follows by close inspection of the structure of the matrix $\bar{B}$, that if $g = 0$, then $\bar{\beta}_{n-d-1} = \bar{\beta}_{n-d} = \ldots = \bar{\beta}_n = 0$ and $\bar{\gamma}_{n-d} = \ldots = \bar{\gamma}_n = 0$ and we can rewrite Equation (2.10) as

$$y(k+n) = \sum_{i=0}^{n-1} \lambda_i y(k+i) + \sum_{i=0}^{n-d} \beta_i u(k+i) \tag{2.12}$$

or, after a time-shift of n instances, as

$$y(k) = \sum_{i=1}^{n} \lambda_{n-i} y(k-i) + \sum_{i=d}^{n} \beta_{n-i} u(k-i). \tag{ARMA}$$

(Note that now the coefficient λ_0 accompanies $y(k-n)$ and so on, thus the smaller i, the "older" the signal accompanied by it. If that is deemed unintuitive the coefficients can be simply renumbered. We will keep the convention that the coefficient with index 0 goes with the oldest value of a signal throughout this text.)

With the derivation of Equation (ARMA) we have obtained another representation of the system in Equation (2.6), namely the so-called *autoregressive moving average (ARMA)* model (the autoregressive part is the first sum, "moving average" refers to the second sum in (ARMA)). Since the state does not appear in that model anymore, it is a input-output (I/O) representation of the system.

The important new insight is, that the output at time k is influenced only by values of the input that are at least d instances of time older than k, so the d we defined Equation (2.11) has a very important interpretation as the *time-delay* of the system.

A very convenient way to write an ARMA model is obtained by introducing the backshift operator q, i.e. $qy(k) = y(k-1)$, $q^n y(k) = y(k-n)$ and so on. Then we can rewrite Equation (ARMA) as

$$A(q)y(k) = q^d B(q)u(k), \tag{2.14}$$

where $A(q) = 1 - \sum_{i=1}^{n} \lambda_{n-i} q^i$ and $B(q) = \sum_{i=d}^{n} \beta_{n-i} q^i$.

2.2.2 The Static Parametric Model (SPM)

Equation (ARMA) can be rewritten as

$$y(k) = \bar{\vartheta}^{*T} \bar{\varphi}(k) \tag{SPM}$$

where

$$\bar{\varphi}(k) := \begin{bmatrix} y(k-n) & \dots & y(k-1) & u(k-n) & \dots & u(k-d) \end{bmatrix}^T \tag{2.16a}$$

$$\bar{\vartheta}^* := \begin{bmatrix} \lambda_0 & \dots & \lambda_{n-1} & \beta_0 & \dots & \beta_{n-d} \end{bmatrix}^T \tag{2.16b}$$

The appeal of that representation stems from the fact that the output (or any linear combination of measurable signals) of the system is connected to a measurable signal $\bar{\varphi} : \mathbb{Z}^+ \to \mathbb{R}^{2n-d+1}$ and the vector of parameters $\bar{\vartheta}^* \in \mathbb{R}^{2n-d+1}$ by a linear static equation. The signal $\bar{\varphi}$ is "assembled" from past values of the known output and input and is called the *regressor* vector. $\bar{\vartheta}^*$ is the *parameter* vector, the asterisk indicates that it is the vector of the true parameters (as opposed to the estimates - refer to the next section). An illustration of that is shown in Figure 2.1. In accordance with e.g. [IF06] we shall call the representation (SPM) *static parametric model (SPM)*.

Remark 2.7 (Known parameters)**.** Another feature of the SPM comes in handy if only some parameters are unknown and need to be estimated. Assume e.g. that λ_{n-r} and β_{n-s} are known (of course, $1 \le r \le n$ and $d \le s \le n$). Then define an auxiliary output $z(k)$, a modified regressor ϕ and a reduced parameter vector θ^* with

$$\begin{aligned} z(k) &:= y(k) - \lambda_{n-r} y(k-r) - \beta_{n-s} u(k-s) \\ \phi(k) &:= \big[\; y(k-n) \quad \dots \quad y(k-r-1) \quad y(k-r+1) \quad \dots \quad y(k-1) \\ &\qquad u(k-n) \quad \dots \quad u(k-s-1) \quad u(k-s+1) \quad \dots \quad u(k-d) \;\big]^T \\ \theta^* &:= \begin{bmatrix} \lambda_0 & \dots & \lambda_{n-r-1} & \lambda_{n-r+1} & \dots & \lambda_{n-1} & \beta_0 & \dots & \beta_{n-s-1} & \beta_{n-s+1} & \dots & \beta_{n-d} \end{bmatrix}^T. \end{aligned}$$

Then the system can be alternatively represented by

$$z(k) = \theta^{*T} \phi(k)$$

and the algorithms can be developed for this model with reduced dimension.

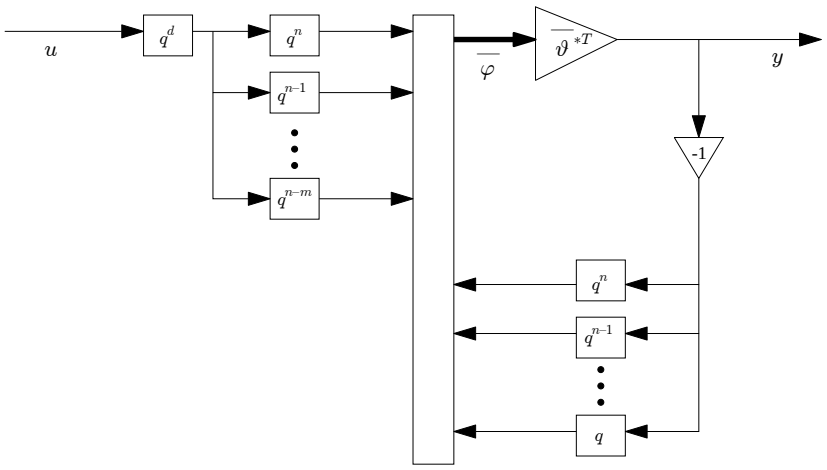

Figure 2.1: A signal-flow diagram of the SPM

2.2.3 The d-step ahead predictor model

Now consider an ARMA model

$$y(k) = \sum_{i=1}^{n} \lambda_{n-i} y(k-i) + \sum_{i=d}^{n} \beta_{n-i} u(k-i) \tag{2.17}$$

with $d \geq 1$. Since the time-delay is d, the inputs used in this model are at least d steps old, but the sequence $u(k-d+1), \ldots, u(k)$ is certainly already known at time k, so the outputs $y(k+1), \ldots, y(k+d)$ are already determined. That gives rise to another possibility to represent the system's behaviour, the *d-step ahead predictor* form

$$y(k+d) = \sum_{i=0}^{n-1} a_{n-1-i} y(k-i) + \sum_{i=0}^{d+n-1} b_{d+n-1-i} u(k-i)^{\downarrow i \downarrow} \tag{DSA}$$

The main characteristic of this form is that there is a "gap" of d steps between the left hand side and the right hand side. In other words, the output only depends on values of the input *and* output that are at least d steps old.

To facilitate the derivation of that result, we again represent the ARMA model with the shift operator q as

$$A(q)y(k) = q^d B(q)u(k), \tag{2.19}$$

where

$$A(q) = 1 + \lambda_{n-1} q + \ldots + \lambda_1 q^{n-1} + \lambda_0 q^n$$
$$B(q) = \beta_{n-d} + \beta_{n-d-1} q + \ldots + \beta_0 q^n.$$

$^{\downarrow i \downarrow}$The weird expression for the indices again is caused by the convention to let the oldest signal value go with the coefficient with index 0.

To achieve sufficient degrees of freedom, we filter both sides by multiplying with a (for the moment) arbitrary polynomial $F(q)$ and q^{-d} to obtain

$$q^{-d}F(q)A(q)y(k) = F(q)B(q)u(k). \tag{2.20}$$

Since $q^{-d}y(k) = y(k+d)$, all we need to do is to get rid of all other negative powers of q on the left hand side, in other words, $F(q)$ should be chosen such that $F(q)A(q) = 1 + q^d G(q)$, so the 1 preserves $q^{-d}y(k)$ and the second part eliminates any other negative power of q. Rewriting this equation we get

$$A(q)F(q) - q^d G(q) = 1. \tag{2.21}$$

Since $A(q)$ is monic, $A(q)$ and q^d are coprime and thus it follows from the Bezout Identity (Theorem A.4) that $F(q)$ and $G(q)$ with

$$\begin{aligned} F(q) &= f_0 + f_1 q + \ldots + f_{d-1} q^{d-1} \\ G(q) &= g_0 + g_1 q + \ldots + g_{n-1} g^{n-1} \end{aligned}$$

exist and we can rewrite Equation (2.20) as $y(k+d) = -G(q)y(k) + F(q)B(q)$ or exactly as in (DSA)

$$y(k+d) = \sum_{i=0}^{n-1} a_{n-1-i} y(k-i) + \sum_{i=0}^{d+n-1} b_{d+n-1-i} u(k-i), \tag{2.22}$$

where $a_{n-1-i} = -g_i$ and $b_{d+n-1-i}$ are the coefficients of the product $F(q)B(q)$ ($b_{d+n-1-i} = \sum_{j+k=i} f_j \beta_k$, where non-existent coefficients of F and B are simply set to zero). After a time shift of d instances, the SPM of this representation can be written as

$$y(k) = \vartheta^{*T} \varphi(k-d) \tag{2.23}$$

where

$$\vartheta^* = \begin{bmatrix} a_0 & a_1 & \ldots & a_{n-1} & b_0 & \ldots & b_{d+n-1} \end{bmatrix}^T \tag{2.24a}$$

$$\varphi(k) = \begin{bmatrix} y(k-n+1) & \ldots & y(k) & u(k-d-n+1) & \ldots & u(k) \end{bmatrix}^T. \tag{2.24b}$$

The signal flow of that representation is depicted in Figure 2.2.

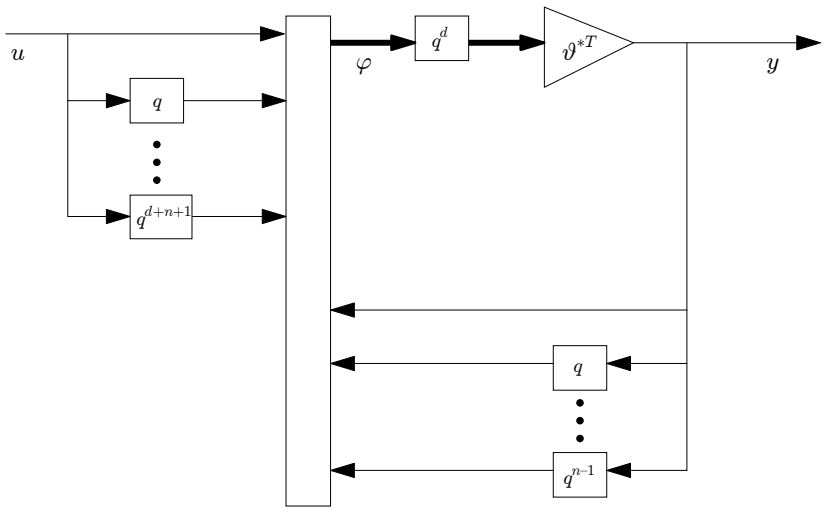

Figure 2.2: The SPM of the d-step ahead predictor model

3 Adaptive Identification and Control of LTI systems

3.1 Adaptive Identification

In this chapter, a short introduction to adaptive identification of a linear I/O behavior is given, using the parametric models that we just introduced, and subsequently extended to the control of such systems if the parameters are unknown.

Unlike in non-adaptive systems identification, where collected I/O data is used to estimate the unknown parameters of a plant, typically using optimization techniques to minimize a certain error functional, adaptive identification makes use of a model whose parameters are dynamically adjusted to make the error between model and plant tend to zero (sometimes this is called "Model Reference Adaptive Identification").

3.1.1 Identification of Static Parametric Models

Problem setup. Let an LTI system be given by its SPM

$$y(k) = \bar{\vartheta^*}^T \bar{\varphi}(k) \tag{3.1}$$

where $\bar{\vartheta^*}$ and $\bar{\varphi}$ as in Equation (2.16), or by its d-step ahead predictor

$$y(k) = \vartheta^{*T} \varphi(k-d), \tag{3.2}$$

where ϑ^* and φ as in Equation (2.24). Note that the structures of (3.1) and (3.2) are identical, save for the time-shift in φ, so any algorithm that we develop for one can also be used for the other. Therefore in the following we will not distinguish between $\bar{\vartheta^*}$ ($\bar{\varphi}$) and ϑ^* (φ) anymore.

Adaptive identification now refers to setting up a model of the plant whose parameter vector is adjusted using available measurements, such that the parameter vector $\vartheta(k)$ tends to the true value ϑ^* asymptotically. Let us use a model structure parallel to the SPM:

$$\hat{y}(k) = \vartheta^T(k-1)\varphi(k) \tag{3.3}$$

where $\vartheta^T(k) = \begin{bmatrix} \hat{\lambda}_0(k) & \dots & \hat{\lambda}_{n-1}(k) & \hat{\beta}_0(k) & \hat{\beta}_{d+n-1}(k) \end{bmatrix}$ and $\hat{\lambda}_i(k), \hat{\beta}_i(k)$ denote the estimates of λ_i and β_i, respectively, at time k. Since the analysis is most conveniently

done in terms of the deviations from the real values, we introduce the identification error $e(k) := \hat{y}(k) - y(k)$ and the parameter error $\tilde{\vartheta}(k) := \vartheta(k) - \vartheta^*$. Combining Equations (3.1) and (3.3), we get

$$e(k) = \vartheta^T(k-1)\varphi(k) - y(k) = \tilde{\vartheta}^T(k-1)\varphi(k). \tag{3.4}$$

To update a parameter, two values have to be chosen: In which direction the parameter should be updated and the size of the step. We will see in the following that it is necessary to normalize the error equation in order to handle the step size, but for the moment we just introduce the normalized identification error ε with no further explanation as

$$\varepsilon(k) = \frac{e(k)}{c + \varphi^T\varphi(k)}, \tag{3.5}$$

where $c > 0$, so the denominator is designed to bound $\|\varphi\|^2$ from above and we can conclude that $\|\varepsilon(k)\varphi(k)\| \leq \frac{\|\varphi(k)\varphi^T(k)\|}{c+\varphi^T\varphi(k)}\|\tilde{\vartheta}(k-1)\| < \|\tilde{\vartheta}(k-1)\|$.
Equation (3.5) together with the adaptive law (yet to be given) which defines the parameter update $\vartheta(k+1) - \vartheta(k) = \tilde{\vartheta}(k+1) - \tilde{\vartheta}(k)$ will constitute the overall system and we will be interested in the stability of the origin ($\varepsilon = 0, \tilde{\vartheta} = 0$).

Remark 3.1. As it will turn out in the course of the following proof of stability, the normalizing signal $n^2(k)$ can be designed arbitrarily, as long as it bounds $\|\varphi\|^2$ from above and fulfills $n^2(k) \geq n_0 > 0\ \forall k \in \mathbb{Z}^+$. If for example it is known beforehand that φ is bounded, a possible choice is $n^2(k) = \sup_{k\in\mathbb{Z}^+} \|\varphi(k)\| > 0$. For the sake of generality we proceed using $n^2(k)$ as a general normalizing signal.

Adaptive Law. A wide class of adaptive laws can be developed by choosing a cost functional and updating the parameter to minimize the cost at every time k. A very simple one would be to update the value of the parameter vector $\vartheta(k)$ to minimize the *instantaneous cost functional*

$$J(\vartheta) = \frac{\varepsilon^2}{2n^2} = \frac{(\vartheta^T\varphi(k) - y(k))^2}{2n^4(k)} \tag{3.6}$$

by the method of the steepest descent, that is $\vartheta(k) = \vartheta(k-1) - \gamma\frac{\partial J}{\partial \vartheta}$ with some gain $\gamma > 0$. Since $\frac{\partial J}{\partial \vartheta} = \varepsilon(k)\varphi(k)$ we get

$$\vartheta(k) = \vartheta(k-1) - \gamma\varepsilon(k)\varphi(k). \tag{3.7}$$

This algorithm has been termed *projection algorithm* in the literature [GS84, IF06]. Although this name originates from another way of deriving this algorithm, we shall use it in the remainder of the thesis. As the following lemma will show, the projection algorithm ensures stability of the overall system.

Lemma 3.2 (Stability of the projection algorithm.)**.** *Consider the overall adaptive system*

$$\varepsilon(k) = \frac{\vartheta^T(k-1)\varphi(k) - y(k)}{n^2(k)} \tag{3.8a}$$

$$\vartheta(k) = \vartheta(k-1) - \gamma\varepsilon(k)\varphi(k) \tag{3.8b}$$

where $n^2(k) \geq \varphi^T(k)\varphi(k)$ *and* $n^2(k) \geq n_0 > 0$ *(since we do not want to divide by 0)* $\forall k \in \mathbb{Z}^+$ *and* $\gamma > 0$ *and the signals* $y : \mathbb{Z}^+ \to \mathbb{R}$ *and* $\varphi : \mathbb{Z} \to \mathbb{R}^{(n+m)}$ *are as defined earlier in the section. Then we can state the following:*
If

$$2 - \gamma\frac{\varphi^T(k)\varphi(k)}{n^2(k)} \geq s_0 > 0 \qquad \forall\, k, \tag{3.9}$$

then

(i) $\vartheta(k) \in \ell^\infty$ *and* $\|\vartheta(k) - \vartheta^*\| \leq \|\vartheta(k-1) - \vartheta^*\| \leq \|\vartheta(0) - \vartheta^*\|$,

(ii) $\varepsilon(k)$, $\varepsilon(k)n(k)$, $\varepsilon(k)\varphi(k)$ *and* $\|\vartheta(k) - \vartheta(k-\nu)\| \in \ell^2$ *for any fixed* $\nu \in \mathbb{Z}^+$, *and*

(iii) $\varepsilon(k)$ *and* $\|\vartheta(k) - \vartheta(k-\nu)\| \to 0$ [↓i↓] *for* $k \to \infty$ *and any fixed* $\nu \in \mathbb{Z}^+$.

Proof. First note, that (ii) ⇒ (iii) since from $a(k) \in \ell^2 \Rightarrow a(k) \to 0$, so we proceed to prove (i) and (ii). From (3.8) we can derive

$$\tilde{\vartheta}(k) = \left[I - \gamma\frac{\varphi\varphi^T}{n^2}(k)\right]\tilde{\vartheta}(k-1) \tag{3.10}$$

which is a linear time-varying system. Now we define the function $V(k) = \gamma^{-1}\tilde{\vartheta}^T(k)\tilde{\vartheta}(k)$[↓ii↓] and find

$$\begin{aligned} V(k) - V(k-1) &= \gamma^{-1}\tilde{\vartheta}^T(k-1)\left[I - \gamma\frac{\varphi\varphi^T}{n^2}(k)\right]^2\tilde{\vartheta}(k-1) - \gamma^{-1}\tilde{\vartheta}^T(k-1)\tilde{\vartheta}(k-1) \\ &= \tilde{\vartheta}^T(k-1)\left[\gamma\frac{\varphi\varphi^T}{n^2}\frac{\varphi\varphi^T}{n^2}(k) - 2\frac{\varphi\varphi^T}{n^2}(k)\right]\tilde{\vartheta}(k-1) \\ &= \varepsilon^2(k)n^2(k)\left[\gamma\frac{\varphi^T\varphi}{n^2}(k) - 2\right]. \end{aligned} \tag{3.11}$$

With (3.9) it follows that $V(k) - V(k-1) \leq -s_0\varepsilon^2(k)n^2(k) \leq 0$ which establishes that $V(k)$ is a monotonically decreasing sequence, which is bounded from below by 0, so the limit $V_\infty = \lim_{k\to\infty} V(k)$ exists and we have

$$s_0 n_0 \sum_{k=2}^{\infty} \varepsilon^2(k) \leq s_0 \sum_{k=2}^{\infty} \varepsilon^2(k)n^2(k) \leq V(1) - V_\infty < \infty \tag{3.12}$$

[↓i↓] This does *not* however imply that $\vartheta(k)$ is converging to any limit. To see that take e.g. a sequence $s(k) = \log(k)$, which fulfills the condition but is unbounded.

[↓ii↓] This function could be interpreted as a Lyapunov function for the time-variant linear system (3.10).

so εn, $\varepsilon \in \ell^2$. With the boundedness of $\frac{\varphi}{n}$ and $\varepsilon\varphi = \varepsilon n \frac{\varphi}{n}$, $\varepsilon\varphi \in \ell^2$ also follows. Furthermore, since $V(k)$ is bounded from above by $V(1)$ and from the definition of $V(k)$ it follows that $\|\tilde{\vartheta}(k)\| \leq \sqrt{\gamma V(k)}$, we can conclude that $\|\tilde{\vartheta}\|$ is monotonically decreasing and bounded and that $\vartheta = \vartheta^* + \tilde{\vartheta}$ is bounded, which is item (i).

The statement $\|\vartheta(k) - \vartheta(k-\nu)\| \in \ell^2$ can be derived by noting that from (3.8) we have

$$\|\vartheta(k) - \vartheta(k-1)\| = \gamma\|\varepsilon(k)\varphi(k)\|, \tag{3.13}$$

so $\varepsilon(k)\varphi(k) \in \ell^2$ ensures that $\|\vartheta(k) - \vartheta(k-1)\| \in \ell^2$, too. Choosing a fixed $\nu \in \mathbb{Z}^+$ we can expand

$$\vartheta(k) - \vartheta(k-\nu) = \vartheta(k) - \vartheta(k-1) + \vartheta(k-1) - \cdots + \vartheta(k-\nu+1) - \vartheta(k-\nu) \tag{3.14}$$

and using the triangle inequality we obtain

$$\begin{aligned}\|\vartheta(k) - \vartheta(k-\nu)\| \leq \|\vartheta(k) - \vartheta(k-1)\| + \|\vartheta(k-1) - \vartheta(k-2)\| + \ldots \\ + \|\vartheta(k-\nu+1) - \vartheta(k-\nu)\|.\end{aligned} \tag{3.15}$$

Since the right hand side obviously consists of ℓ^2 sequences, $\|\vartheta(k) - \vartheta(k-\nu)\| \in \ell^2$, too, what concludes the proof. □

Some remarks seem in order at this point:
What we have proven so far is merely that the adaptive law ensures that the *normalized* identification error converges to zero and that the next parameter estimate never lies farther away from the real value ϑ^* than the current one. As noted in the footnote to Lemma 3.2, this does not imply that the parameter errors converge to any constant, especially not to zero, so we have not yet said anything about the actual goal of the whole procedure, the convergence of $\vartheta(k)$ to ϑ^*. Furthermore, since not even the boundedness of φ was assumed, the convergence of ε does not imply that the unnormalized identification error e is bounded, since φ could grow at a faster rate than e.

Pure identification of unstable systems however does not make much sense, so the boundedness of φ can be assumed in this case. Problems will occur later when an identification algorithm is teamed with a control algorithm and stability of the plant cannot be assumed anymore.

But in the remainder of this section we shall confine our attention to stable systems and convergence of the parameter estimates to their real values. It is clear that this depends on the chosen input u: If for instance the input u is equal to zero, it is obvious that the output of the system as well as the output error carry no information about the parameters β_i or b_i and thus no adaptive scheme could ensure convergence of ϑ to ϑ^*. So intuitively one could think that, since the only information about the success of the identification we have is the identification error, one has to make sure that the error contains enough information about the uncertain parameters to make parameter convergence possible in the first place. This notion is formalized in the following lemma.

Lemma 3.3 (Parameter convergence)**.** *If all conditions of Lemma 3.2 hold, and additionally there exist* $t \in \mathbb{Z}^+$ *and* $\delta > 0$ *such that*

$$\sum_{i=0}^{t-1} \frac{\varphi\varphi^T}{n^2}(k+i) \geq \delta I \qquad \forall\, k \in \mathbb{Z}^+, \tag{3.16}$$

then $\lim_{k\to\infty} \vartheta(k) = \vartheta^*$.

Proof. The proof basically consists of showing that under condition (3.16), the time-varying system (3.10) is uniformly completely observable. Since this procedure requires some theorems and techniques that are of no interest for the remainder of this thesis, the proof is omitted and can be found in e.g. [IF06, GS84]. □

Definition 3.4 (PE)**.** Signals $\varphi : \mathbb{Z}^+ \to \mathbb{R}^n$ that fulfill condition (3.16) are called *persistently exciting (PE).*

Condition (3.16) is a condition on the regressor vector φ, the only part that can be designed directly however is the input u. The results how to choose u such that φ is PE are roughly intuitive but the details are immensely complex, this is why we do not give a lemma or theorem but again refer the reader to [IF06, GS84] and merely state as a rule-of-thumb:

> If the system to be identified is stable and ν parameters need to be estimated by an adaptive law, then an input of the form
>
> $$u(k) = \sum_{i=1}^{\lceil \nu/2 \rceil} c_i \sin(\omega_i k),$$
>
> where $\lceil x \rceil = \min\{n \in \mathbb{N} |\, n \geq x\}$ and $c_i, \omega_i \neq 0\ \forall i$, $\omega_i \neq \omega_j$ if $i \neq j$, ensures parameter convergence.

3.1.2 Simulation: Adaptive Identification of a second order system

We now digress briefly to illustrate the findings so far with a simple example. Let a second order LTI system be given by

$$y(k+2) = \lambda_1 y(k+1) + \lambda_0 y(k) + \beta_0 u(k) \tag{3.17}$$

with $y(1) = y_0$ and $y(2) = y_1$ and where the λ_i and $\beta_0 \neq 0$ are unknown but constant and such that the system is stable. The SPM of this plant is

$$y(k) = \bar{\vartheta^*}^T \bar{\varphi}(k) \tag{3.18}$$

where

$$\bar{\vartheta}^* = \begin{bmatrix}\lambda_0 & \lambda_1 & \beta_0\end{bmatrix}^T$$
$$\bar{\varphi}(k) = \begin{bmatrix}y(k-2) & y(k-1) & u(k-2)\end{bmatrix}^T.$$

The time-delay of the plant obviously is $d = 2$ and so we can also derive the 2-step ahead predictor model of the plant by expressing $y(k+1)$ as

$$y(k+1) = \lambda_1 y(k) + \lambda_0 y(k-1) + \beta_0 u(k-1) \tag{3.19}$$

and inserting into (3.17) to obtain

$$y(k+2) = a_1 y(k) + a_0 y(k-1) + b_1 u(k) + b_0 u(k-1), \tag{3.20}$$

where $a_1 = \lambda_1^2 + \lambda_0$, $a_0 = \lambda_1\lambda_0$, $b_1 = \beta_0$ and $b_0 = \lambda_1\beta_0$, or

$$y(k) = \vartheta^{*T}\varphi(k-2), \tag{3.21}$$

where

$$\vartheta^* = \begin{bmatrix}a_0 & a_1 & b_0 & b_1\end{bmatrix}^T$$
$$\varphi(k) = \begin{bmatrix}y(k-1) & y(k) & u(k-1) & u(k)\end{bmatrix}^T.$$

Following the developments of the last section, the SPM (3.18) as well as the 2-step ahead predictor (3.21) were identified. The values of the various parameters thereby were:

$$\lambda_1 = \frac{3}{4} \qquad \lambda_0 = -\frac{1}{8}$$
$$\beta_0 = 1$$

and thus

$$a_1 = \frac{7}{16} \qquad a_0 = -\frac{3}{32}$$
$$b_1 = 1 \qquad b_0 = \frac{3}{4}.$$

Furthermore, $n^2(k) = 1 + \varphi^T(k)\varphi(k)$ and $\gamma = 1.5$ and the initial conditions were $y(1) = y(0) = 0$, $\hat{b}_i(1) = \hat{\lambda}_i(1) = \hat{a}_1(1) = \hat{\beta}_0(1) = 1$, $i = 0, 1$, and $\hat{a}_0(1) = 2$. To illustrate the necessity of the PE condition (3.16), two different inputs were used, first $u_1(k) = 10$, which is not PE, and then $u_2(k) = 10 + 20\sin(2k) + 30\sin(k)$ which is. The results are shown in Figures 3.1, 3.2, 3.3 and 3.4. As expected, the identification error converges to zero in all cases, whereas the parameters converge to constant but wrong values where the input signal is a constant (Note that in Figure 3.2, the lines for $\hat{b}_0$ and $\hat{b}_1$ coincide, which is no surprise since the last two entries in $\bar{\varphi}$ are the same and thus $\hat{b}_0(k) - \hat{b}_0(k-1) = \hat{b}_1(k) - \hat{b}_1(k-1)$ for all k.) and to their true values only if the input is PE.

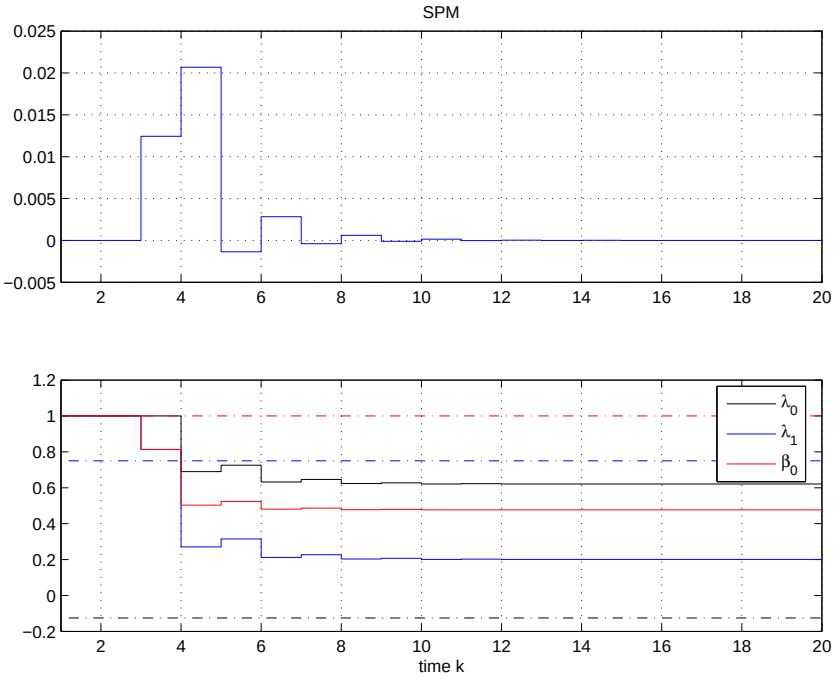

Figure 3.1: Identification of the SPM using a constant input signal

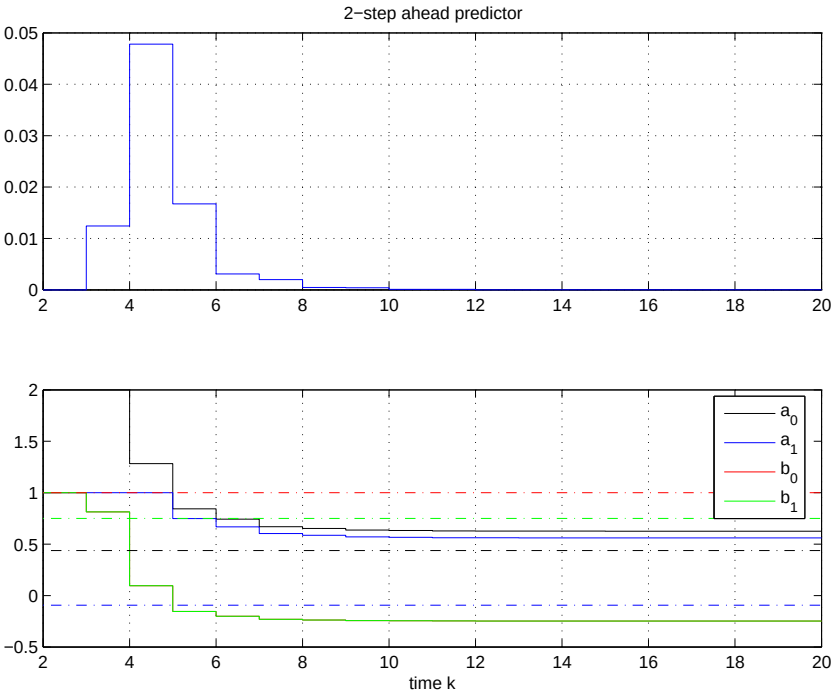

Figure 3.2: Identification of the 2-step ahead predictor using a constant input signal

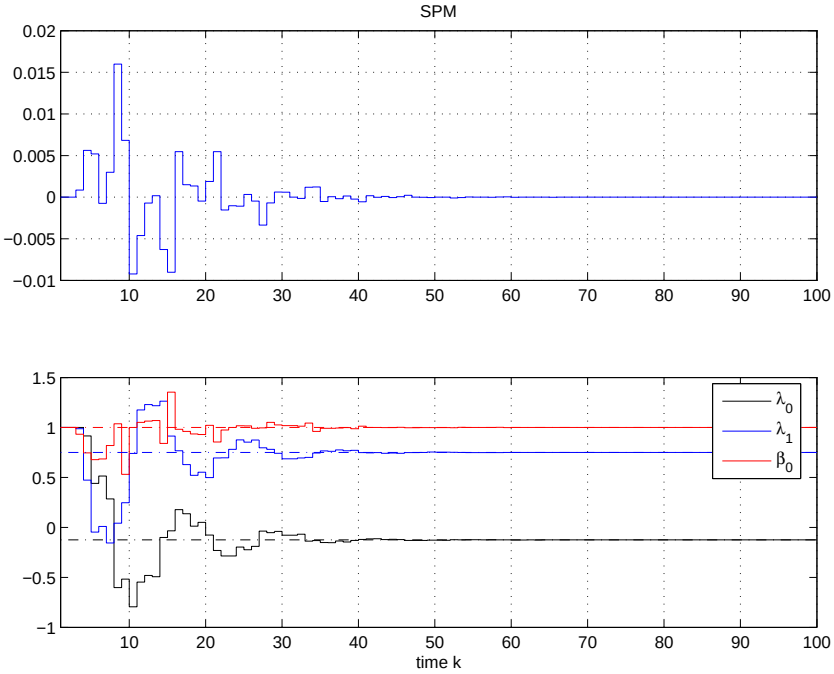

Figure 3.3: Identification of the SPM using a persistently exciting input signal

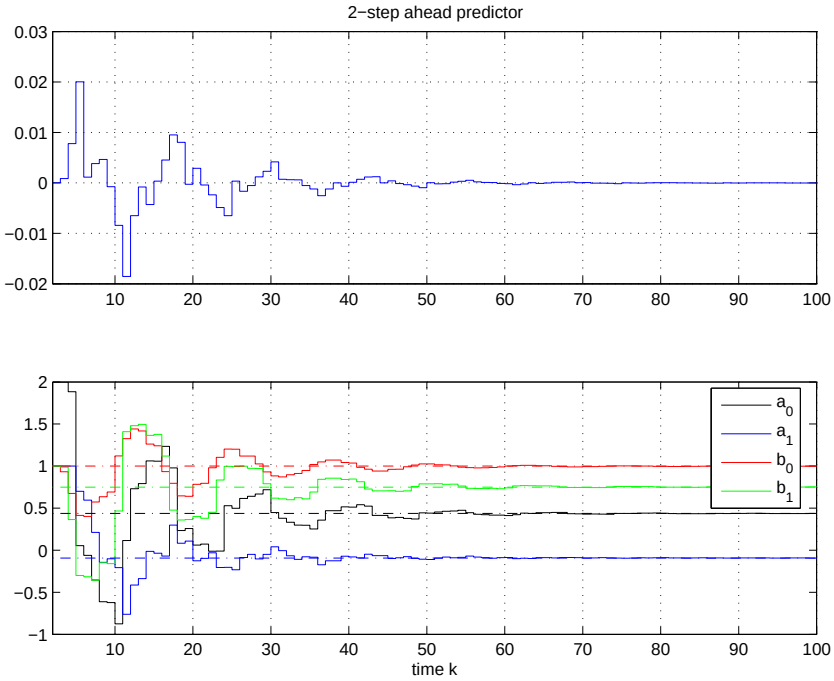

Figure 3.4: Identification of the 2-step ahead predictor using a persistently exciting input signal

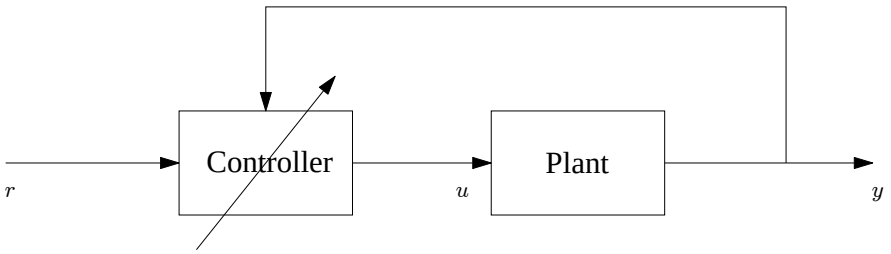

Figure 3.5: The direct approach

3.2 Adaptive Control

Now that algorithms to estimate the unknown parameters of a plant have been introduced, the next step is to expand this concept to adaptive *control* of unknown plants. Two different philosophies can be followed: In the *direct* approach which is depicted in Figure 3.5, the parameters of the controller are dynamically adjusted based on the available measurements to achieve a control objective. In the *indirect* approach which is depicted in Figure 3.6, the parameters of a plant model are dynamically adjusted (i.e. the plant is identified) and the control parameters are then computed from the estimates with an algebraic relation. While both approaches have received wide interest and application in the literature, in view of the scope of this thesis we will restrict ourselves to the indirect approach: Identification does not affect the plant in that it does not close any control loops between plant output and input, so arbitrarily many independent identifiers can be connected to the plant and tuned separately whereas in the direct approach, the tuning of the parameters relies on the control error, that is the difference of reference signal and plant output what in turn means that only the performance of the active controller can be evaluated and consequently only one controller can be tuned at a time. If that was not clear now, it will become clear later when the Multiple-Model approaches are introduced.

So as a next step, a simple control algorithm will be introduced for the non-adaptive case and then it will be shown how this control algorithm can be combined with a stable identification algorithm to obtain an adaptive controller. After the proof of stability, an example will conclude this section.

3.2.1 One-step-ahead control

Our starting point is an ARMA model

$$A(q)y(k) = B(q)u(k) \tag{3.22}$$

of a linear time invariant plant. We have seen in Section 2.2 that we can derive its d-step ahead predictor form as

$$y(k+d) = \vartheta^{*T}\varphi(k), \tag{3.23}$$

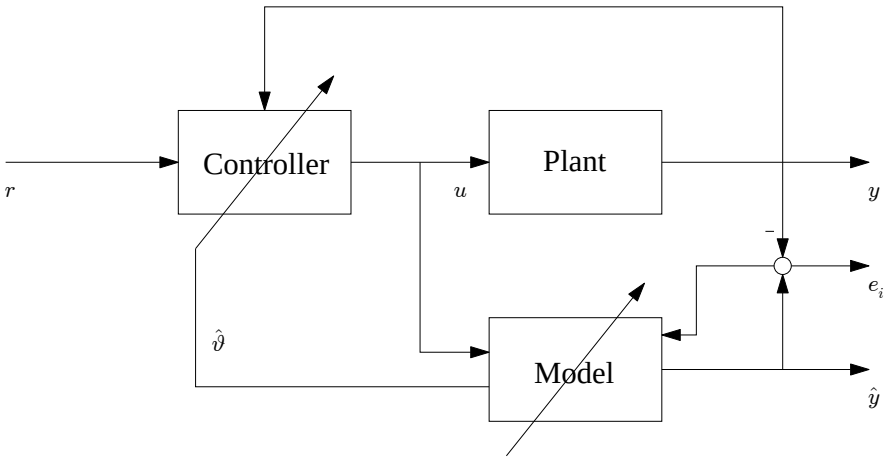

Figure 3.6: The indirect approach

where the parameter vector ϑ^* and the regressor φ are given by Equation (2.24) and $b_{d+n-1} \neq 0$, provided that we chose the "right" delay d as in Equation (2.11). The control objective is to have the output $y(k)$ track the *reference signal* $y^*(k)$.

At a given time k, the outputs $y(k), y(k-1), \ldots$ as well as the past inputs $u(k-1), u(k-2), \ldots$ are known and can be used to compute the current input $u(k)$ such that ${\vartheta^*}^T \varphi(k) = y^*(k+d)$ by simply inverting this relationship to obtain

$$u(k) = \frac{1}{b_{d+n-1}} \left(y^*(k+d) - a_{n-1} y(k) - a_{n-2} y(k-1) - \cdots - b_{d+n-2} u(k-1) - \cdots \right) \tag{3.24}$$

(note that we assumed $b_{d+n-1} \neq 0$). A note on the initial conditions: The order of the system is n, that is n initial conditions are needed, which in turn means that $y(1), \ldots, y(n)$ are fixed and the actual simulation and prediction does not start until $k = n$. The first value that can be predicted by the d-step ahead predictor thus is $y(n+d)$. So it does not make sense to use the feedback-law (3.24) for the first $n-1$ time-steps. The input sequence $u(1), \ldots, u(n-1)$ can be chosen arbitrarily to compute $u(n)$ at time n[iii]. So the construction of the feedback-law (3.24) ensures that

$$y(k) = y^*(k) \qquad \forall\, k \geq n + d. \tag{3.25}$$

The next question we have to answer is whether the control law results in a bounded input signal. Inserting (3.25) into (3.22) yields

$$B(q)u(k) = A(q)y^*(k) \qquad \forall k \geq n + d, \tag{3.26}$$

that is u is bounded if and only if this "inverse model" is BIBO stable. A sufficient condition is that all roots of $B(z^{-1})$ lie inside the unit circle, for details see Lemma A.3.

[iii] Of course if the initial values $y(1), \ldots, y(n)$ are known, the initial input sequence $u(1), \ldots, u(n)$ can be chosen in a more useful way, e.g. to minimize the effort $\left[\begin{smallmatrix} u(1) & \ldots & u(n) \end{smallmatrix}\right]^T \left[\begin{smallmatrix} u(1) & \ldots & u(n) \end{smallmatrix}\right]$.

It should be noted here, that this is not an algorithm for direct practical application since it suffers from many drawbacks like possibly huge input amplitudes and non-robustness. Since this thesis is more about showing some new ways than about presenting ready-to-use algorithms, we will leave it with that and refer the interested reader to e.g. [GS84, Ch. 5], where several modifications to overcome some of those difficulties can be found.

3.2.2 Adaptive one-step ahead control

Now we proceed to utilize the one-step ahead control algorithm to obtain an adaptive control algorithm. The idea is fairly simple: If the parameters of the ARMA model (3.22) or the d-step ahead predictor (3.23) are not known exactly, the control law (3.24) cannot be implemented as is. Instead, the identification techniques of Section 3.1.1 are used to obtain an estimate $\vartheta(k)$ of the parameter vector ϑ^* and the estimate is used at every time k to compute the input according to (3.24). Using for instance algorithm (3.7), the complete control algorithm is: At every time $k \geq n$ when $y(k)$ becomes available, do

$$\varepsilon(k) = \frac{\vartheta^T(k-1)\varphi(k-d) - y(k)}{n^2(k-d)} \tag{3.27a}$$

$$\vartheta(k) = \vartheta(k-1) - \gamma\varepsilon(k)\varphi(k-d) \tag{3.27b}$$

$$u(k) = \frac{y^*(k+d) - \hat{a}_{n-1}(k)y(k) - \cdots - \hat{b}_{d+n-2}(k)u(k-1) - \dots}{\hat{b}_{d+n-1}(k)} \tag{3.27c}$$

$$\varphi(k) = \begin{bmatrix} y(k-n+1) & \dots & y(k) & u(k-d-n+1) & \dots & u(k) \end{bmatrix}^T, \tag{3.27d}$$

where $\vartheta(k) = \begin{bmatrix} \hat{a}_0(k) & \dots & b_0(k) & \dots \end{bmatrix}^T$, $\hat{a}_i$ and $\hat{b}_i$ denote the parameter estimates as usual and $\gamma > 0$ and n^2 are chosen such that the conditions of Lemma 3.2 hold.

Equations (3.27a) and (3.27b) constitute the identification part of the algorithm, (3.27c) is the control law and (3.27d) is merely the update of the regressor vector.

In view of Equation (3.27c) we also have to ensure, that $\hat{b}_{d+n-1}(k) \neq 0 \ \forall k$. If the delay d is correct, i.e. is known exactly, a very simple way to do so is to add the assumption, that the sign and a lower bound $\underline{b} > 0$ on the absolute value of b_{d+n-1} is known and to modify the identification algorithm with

$$\vartheta'(k) = \vartheta(k-1) - \gamma\varepsilon(k)\varphi(k-d)$$

$$\vartheta(k) = \begin{cases} \vartheta'(k) & \text{if } \operatorname{sign}(b_{d+n-1})\hat{b}_{d+n-1}(k) \geq \underline{b} \\ \begin{bmatrix} \hat{a}_0(k) & \dots & \hat{b}_{d+n-2}(k) & \operatorname{sign}(b_{d+n-1})\underline{b} \end{bmatrix}^T & \text{else.} \end{cases}$$

that is simply setting $\hat{b}_{d+n-1}(k)$ to the bound if it is too close to zero.

As a last difficulty, if the system is to follow arbitrary reference signals $y^*(k)$ it is impossible to guarantee for $\varphi(k)$ to be PE, i.e. fulfill (3.16), so the parameter estimate $\vartheta(k)$

will probably not converge to ϑ^*. Does it make any sense to base control on "wrong" parameters, will this yield a stable close-loop system? As it will turn out in the remainder of this section, the answer to both questions is yes.

To get started with the analysis, we introduce the *control error* e_c and the *normalized control error* ε_c with

$$\begin{aligned} e_c(k) &:= y(k) - y^*(k) \\ \varepsilon_c(k) &:= \frac{e_c(k)}{c + \varphi^T \varphi(k-d)}, \end{aligned} \tag{3.28}$$

where $c > 0$.

We first collect the assumptions made in the prequel and state in the following lemma that the normalized control error goes to zero and that the conditions, under which that implies that the control error goes to zero as well, are always fulfilled.

Assumptions 3.5.

(i) The delay d is known exactly.

(ii) The sign of b_{d+n-1} and a lower bound $\underline{b}$ for $|b_{d+n-1}|$ are known.

(iii) The order n, or at least an upper bound $\overline{n} \geq n$ for the order n, is known.[↓iv↓]

(iv) The inverse system $B(q)z(k) = A(q)v(k)$ is BIBO stable (cf. Lemma A.3).

Lemma 3.6 (Indirect adaptive control)**.** *Assume a linear plant of the form (3.22) and that Assumptions 4.1 are fulfilled. Then, the adaptive one-step ahead control law (3.27c) together with any stable identification scheme guarantees, that $u \in \ell^\infty$ and $e_c \in \ell^2$ (which implies $e_c(k) \to 0$ as $k \to \infty$).*

Proof. The proof is given in 3 steps: First, we show that $\varepsilon_c \to 0$. Then, the condition under which that also implies $e_c \to 0$ is given. The proof is concluded in the third step by showing that this condition is always fulfilled in the considered case. Boundedness of u follows as a byproduct from the boundedness of $\|\varphi\|$.

1. $\varepsilon_c \to 0$ and $\varepsilon_c \sqrt{c + \varphi^T \varphi} \to 0$.

   Simply observe, that

$$\begin{aligned} \varepsilon_c(k) &= \frac{\vartheta^{*T} - \vartheta^T(k-d)}{c + \varphi^T \varphi(k-d)} \varphi(k-d) \\ &= \frac{\varphi^T(k+d)}{c + \varphi^T \varphi(k-d)} \left[\vartheta^* - \vartheta(k-1) + \vartheta(k-1) - \vartheta(k-d)\right] \\ &= -\varepsilon(k) + \frac{\varphi^T(k-d)}{c + \varphi^T \varphi(k-d)} \left[\vartheta(k-1) - \vartheta(k-d)\right] \end{aligned}$$

[↓iv↓] If the model is overparametrized, the spurious coefficients simply tend to zero. For ease of notation we proceed by assuming n is known, otherwise simply substitute n with $\overline{n}$.

and from that and the triangle equality

$$|\varepsilon_c(k)| \leq |\varepsilon(k)| + \frac{\|\varphi^T(k-d)\|}{c+\varphi^T\varphi(k-d)} \|\vartheta(k-1) - \vartheta(k-d)\| . \tag{3.29}$$

Since $\frac{\|\varphi^T(k-d)\|}{c+\varphi^T\varphi(k-d)}$ is bounded and from Lemma 3.2 we have that $\varepsilon(k), \varepsilon\sqrt{c+\varphi^T\varphi}$, and $\|\vartheta(k-1) - \vartheta(k-d)\| \in \ell^2$. $\varepsilon_c \in \ell^2$ and (by multiplying (3.29) with $\sqrt{c+\varphi^T\varphi}$) $\varepsilon_c\sqrt{c+\varphi^T\varphi} \in \ell^2$ follow.

2. From 1. and $\|\varphi(k-d)\| \leq D_1 + D_2 \max_{\tau\leq k} |e_c(\tau)| \Rightarrow e_c \to 0$.

   First note that if the condition holds and in addition $e_c \in \ell^\infty$, $\|\varphi\| \in \ell^\infty$ directly follows. Then let D_3 be the upper bound on $\|\varphi\|$ and we have $\varepsilon_c(k) \geq \frac{e_c(k)}{c+D_3^2}$, or $e_c(k) \leq (c+D_3^2)\varepsilon_c(k)$ and thus $e_c(k) \to 0$.
   So all that remains to show is that 1. and the condition on $\|\varphi\|$ already imply that $e_c(k)$ is bounded. The proof will be given by contradiction: Assume that $e_c(k)$ grows beyond every bound. Then we can choose an infinite sequence $\{\kappa_1, \kappa_2, \ldots\} \subset \mathbb{Z}^+$ such that

   $$|e_c(\kappa_{i+1})| > |e_c(k)| \qquad \forall k < \kappa_{i+1} \tag{3.30}$$

   (otherwise $|e_c(\kappa_i)|$ is an upper bound) and thus $e_c(\kappa_i) \to \infty$. If we now observe $\varepsilon_c\sqrt{c+\varphi^T\varphi}$ along the sequence κ_i, we find

   $$\begin{aligned} |\varepsilon_c(\kappa_i)\sqrt{c+\varphi^T\varphi}| &\geq \frac{|e_c(\kappa_i)|}{\sqrt{c+\varphi^T\varphi(\kappa_i-d)}} \\ &\geq \frac{|e_c(\kappa_i)|}{\sqrt{c}+\|\varphi(\kappa_i-d)\|} && \text{because } \sqrt{\cdot} \text{ is concave} \\ &\geq \frac{|e_c(\kappa_i)|}{\sqrt{c}+D_1+D_2|e_c(\kappa_i)|} && \text{from (3.30).} \end{aligned}$$

   So we have that $\lim_{\kappa_i\to\infty} |\varepsilon_c(\kappa_i)\sqrt{c+\varphi^T\varphi}| \geq \frac{1}{D_2} > 0$, clearly a contradiction, since every subsequence of the convergent sequence $|\varepsilon_c\sqrt{c+\varphi^T\varphi}|$ has to converge to the same limit, namely 0. So the assumption must have been wrong and e_c is bounded.

3. $\|\varphi(k-d)\| \leq D_1 + D_2 \max_{\tau\leq k} |e_c(\tau)|$ holds.

   From the assumption, that the inverse model is stable, it follows, that constants $D_4, D_5' > 0$ exist, such that $|u(k-d)| \leq D_4 + D_5' \max_{\tau\leq k} |y(k)|$[↓v↓] Let $D_5 = \max\{1, D_5'\}$ and p be the dimension of φ. Then:

   $$\begin{aligned} \|\varphi(k-d)\| &\leq p\,[\max|\varphi_i(k-d)|] \\ &\leq p \max_{\tau\leq k}\{D_4 + D_5'|y(\tau)|, |y(\tau)|\} \\ &\leq p\Big[D_4 + D_5 \max_{\tau\leq k}|y(\tau)|\Big] \end{aligned}$$

[↓v↓] Since we assumed the inverse system $q^d B'(q)z(k) = A(q)v(k)$ to be stable, it is a well known result from systems theory, that the system is finite gain stable, i.e. $\exists A_1, A_2 : |z(k-d)| \leq A_1 + A_2 \max_{\tau\leq k} |v(\tau)|$.

and from the definition of e_c we have $|y| \leq |e_c| + |y^*| \leq |e_c| + D_6$ (where D_6 is the upper bound to y^*), so we can conclude:

$$\|\varphi(k-d)\| \leq D_1 + D_2 \max_{\tau \leq k} |e_c(\tau)|,$$

where $D_1 = p(D_4 + D_5 D_6)$ and $D_2 = pD_5$, what concludes the proof.

□

From the course of the proof it turns out that the notion of "stable" identification algorithm can be refined to "an algorithm which ensures that $\varepsilon, \|\vartheta(k) - \vartheta(k-N)\| \in \ell^2$", so it especially is not necessary, that the parameter vector converges to the real value ϑ^*.

3.2.3 Simulation: Adaptive Control of a 3rd order system

Like Section 3.1, we conclude this Section with a simple example demonstrating the application of the results given. The starting point again is an ARMA model with input delay $d = 2$:

$$y(k+3) = \sum_{i=0}^{2} \lambda_i y(k+i) + \sum_{i=0}^{1} \beta_i u(k+i) = \bar{\vartheta^*}^T \bar{\varphi}(k+2), \tag{3.31}$$

where $\bar{\vartheta^*} = \begin{bmatrix} \lambda_0 & \lambda_1 & \lambda_2 & \beta_0 & \beta_1 \end{bmatrix}^T$ and $\bar{\varphi}(k) = \begin{bmatrix} y(k-2) \cdots & u(k-2) & \cdots \end{bmatrix}^T$. For known parameters λ_i, β_i, the 2-step ahead predictor form could be computed and used for control like proposed in Section 3.2.1. We however assume that the parameters are unknown, so we can only derive the structure of the 2-step predictor to be

$$y(k) = \sum_{i=0}^{2} (a_i y(k-i-2) + b_i u(k-i-2)) =: \vartheta^{*T} \varphi(k-2) \qquad k \geq 4,$$

where $\vartheta^* = \begin{bmatrix} a_0 & \cdots & b_0 & \cdots \end{bmatrix}^T$ and $\varphi(k) = \begin{bmatrix} y(k-2) & \cdots & u(k-2) & \cdots u(k) \end{bmatrix}^T$ as usual, and use this structure to set up the indirect adaptive control algorithm like in Equation (3.27).

The parameter values were chosen to be the following:

$$\lambda_2 = \frac{3}{4} \qquad \lambda_1 = -2$$
$$\lambda_0 = \frac{5}{2}$$
$$\beta_1 = 1 \qquad \beta_0 = \frac{1}{2},$$

so the system is unstable with a pole at $p = \frac{3}{2}$, but the inverse system is stable. The reference signal y^* is a pulse train

$$y^*(k) = \begin{cases} 5 & \text{if} \quad n \cdot 100 \leq k < n \cdot 100 + 50 \\ 2 & \text{if} \quad n \cdot 100 + 50 \leq k < n \cdot 100 \end{cases},$$

$\gamma = 1$ and the initial values are $y(1) = y(2) = y(3) = 1$ and

$$\vartheta(3) = \vartheta^* \bullet \begin{bmatrix} 1.1 & 1.05 & 0.9 & 0.95 & 0.9 & 1.1 \end{bmatrix}^T,$$

where $\bullet$ denotes the element-wise or Hadamard product, that is the numbers in the second vector denote the multiplicative error of the initial estimates. The simulation results are shown in Figure 3.7. It can be seen that after an initial "training" phase with large transients, the controller performance is quite convincing although most parameters, as expected, do not converge to their true values. Also the aforementioned disadvantage of large control amplitudes is clearly visible.

This example concludes the section on classical "single model" adaptive control and identification as well as the introductory chapter. In the next chapter, a short overview of the Multiple-Model approach is presented, before we move on to the announced "new method".

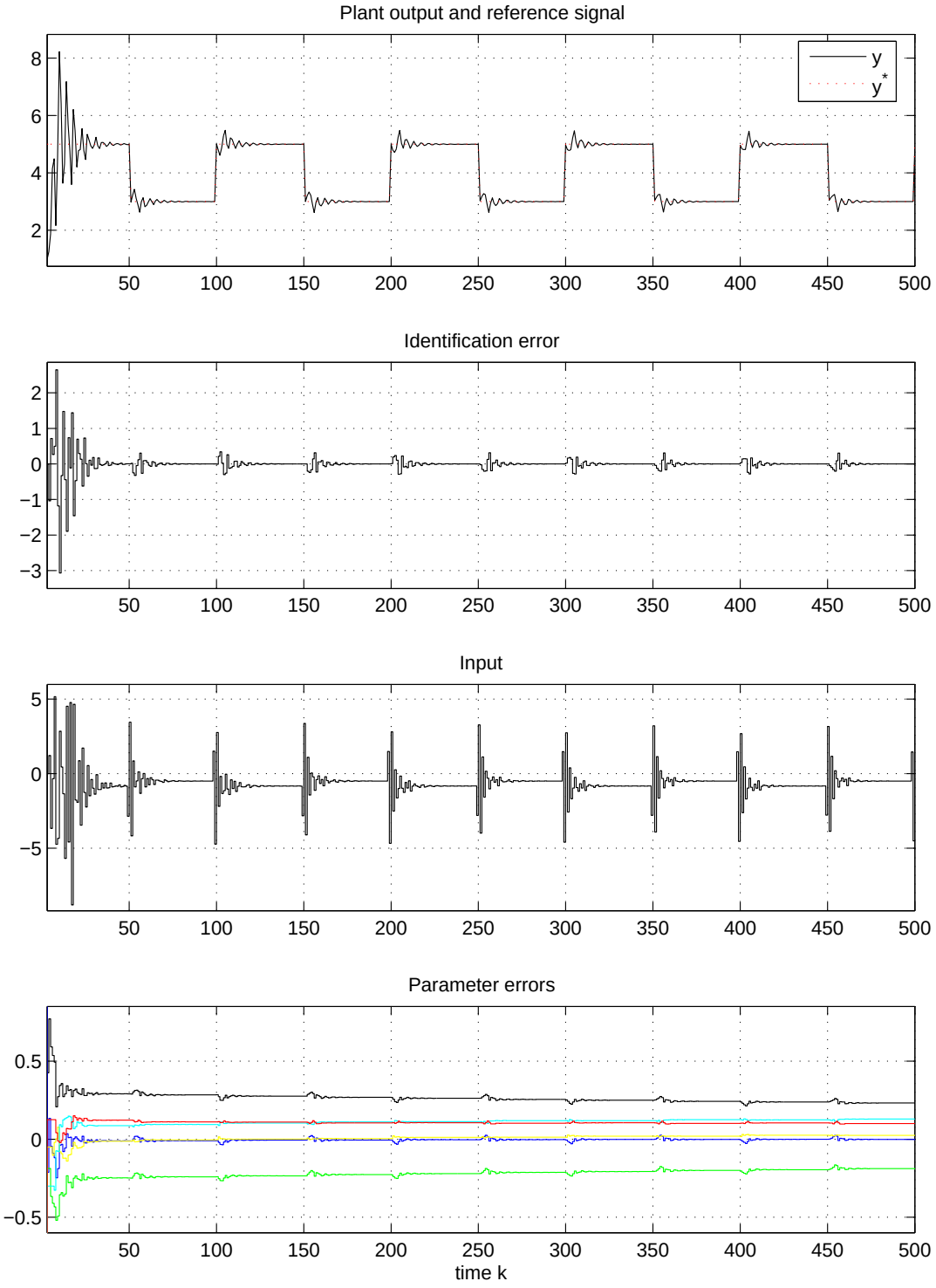

Figure 3.7: Indirect adaptive control of an unstable linear plant

4 Adaptive Identification and Control using multiple models

This chapter is meant to provide a short overview of the developments in Adaptive Control with multiple models which up to now is the main method used to handle rapid and large time variations of the parameters of the controlled plant. First, the necessity for these approaches is shown with two simple examples. Second, the developments in that area are summarized and the dominant approach involving switching and tuning is outlined. Finally, simulations are presented which show that the approach indeed improves the performance of the system in the cases used for the introductory examples.

4.1 Introductory examples

The adaptive schemes treated so far as well as other schemes that fall in the same era of adaptive control were found to perform well - possibly with some modifications to improve robustness, speed or accuracy - if two conditions on the controlled system are satisfied:

- ▷ The initial parameter error $\|\tilde{\vartheta}(0)\|$ is small enough in some sense, and
- ▷ the parameters are constant over time or change on a considerably slower time-scale than the adaptation.

Otherwise the performance of the overall system can be degraded by huge transients or, in the case of time variations, even instability. To illustrate this, the simulations from Section 3.2 are repeated with modification of the aforementioned environmental parameters.

Large initial parameter error: Large transients. This simulation uses the exact same setup as in Section 3.2.3, this time with the reference signal $y^*(k) = 10\sin(k/10)$. The simulation is run two times: The first time, the initial parameter vector is

$$\vartheta(3) = \vartheta^* \bullet \begin{bmatrix} 1.1 & 1.05 & 0.9 & 0.95 & 0.9 & 1.1 \end{bmatrix}^T$$

like in 3.2.3, the second time the initial errors are considerably larger, namely

$$\vartheta(3) = \vartheta^* \bullet \begin{bmatrix} 1.3 & 1.3 & 0.8 & 0.7 & 0.8 & 1.3 \end{bmatrix}^T .$$

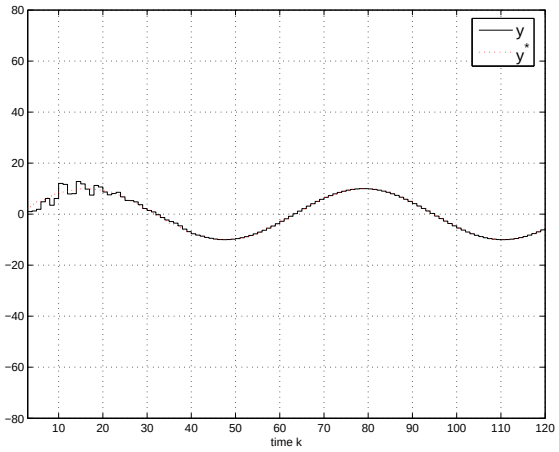

Figure 4.1: A small initial parameter error causes only reasonable transients in the start-up phase

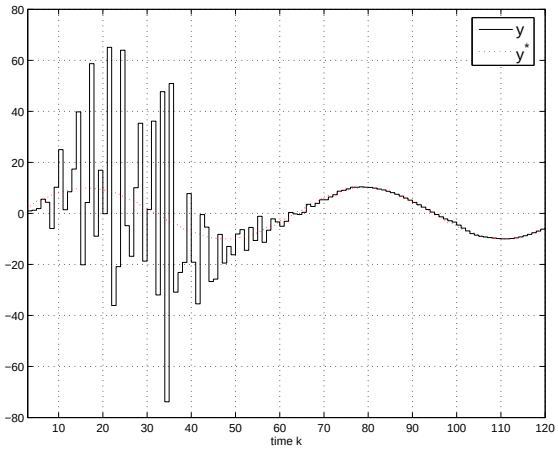

Figure 4.2: A large initial parameter error causes severe degradation of the initial performance

The results are shown in Figures 4.1 and 4.2. It is obvious that the transients caused by the large initial error are hardly tolerable in any practical application, even more so since in the transient phase the controller also calls for large control amplitudes.

Time-variations: Large transients and instability. Again, the same system and algorithms as in Section 3.2.3 and the reference signal $y^*(k) = 10\sin(k/10)$ are used, but this time, the parameter vector of the system changes every 100 instants of time, so $\bar{\vartheta}^*(k) = \bar{\vartheta}^*_{\lceil k/100 \rceil \mod 4+1}$. The parameter values used thereby are

$$\begin{aligned} \bar{\vartheta}^*_1 &= \begin{bmatrix} 5/2 & -2 & 3/4 & 1 & 1/2 \end{bmatrix} \\ \bar{\vartheta}^*_2 &= \begin{bmatrix} -1 & 1 & 1/2 & 2 & 1 \end{bmatrix} \\ \bar{\vartheta}^*_3 &= \begin{bmatrix} 1/2 & 2 & -2 & 1/2 & 1/3 \end{bmatrix} \\ \bar{\vartheta}^*_4 &= \begin{bmatrix} -1 & -1 & 1 & 2 & -1 \end{bmatrix}. \end{aligned} \tag{4.1}$$

Time variations like this could very well be the effect of failure in system components or discrete changes in the environment, e.g. transition between different working modes of a chemical reactor.
Note, that this example is more or less a mere extension of the one before: The problems are caused by the large transients which in turn occur because whenever the parameter vector changes that is equivalent to initiating adaptation with a possibly very large initial parameter error.
The simulation results if the parameter estimate is initialized with

$$\vartheta(3) = \vartheta^* \bullet \begin{bmatrix} 1.1 & 1.05 & 0.9 & 0.95 & 0.9 & 1.1 \end{bmatrix}^T$$

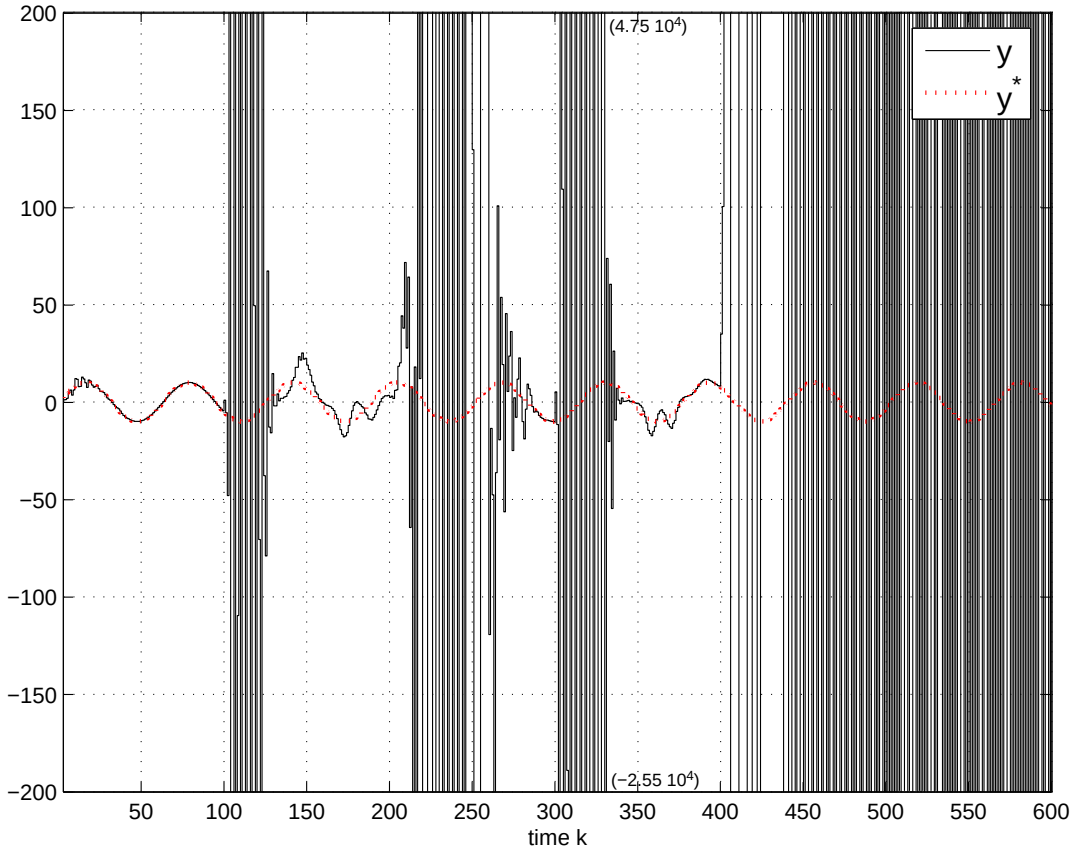

Figure 4.3: Fast time-variations can lead to large transients and instability. The spikes are cut off to show that the adaptive controller is able to stabilize the plant after a certain amount of time during the first few intervals.

are shown in Figure 4.3. As expected, the performance is severely degraded, including instability from the 4th switching instance ($k = 400$) on.

Efforts to overcome this limitations during the late 1980s and 1990s focussed on improving the speed of the adaptive schemes to attenuate transient effects and prevent instability. The concept of *Multiple Models, Switching and Tuning* thereby can be seen as the now dominant approach and so this chapter is devoted to introducing and sketching the scheme after giving a short historical overview of the developments. As usual, simulations illustrating the effectivity of the proposed algorithms will conclude the chapter.

4.2 Multiple-Model approaches in control

4.2.1 Historical overview

The use of multiple models in identification and control can be traced back to the 1960s and 1970s, when in the works of e.g. Magill [Mag65], Saridis and Dao [SD72], Lainiotis [Lai76] and Athans et. al. [ACKP+77] theory and application of the so-called MMAC (Multiple Model Adaptive Control) approach were reported. These early approaches typically consist of a bank of Kalman Filters designed for different possible values of the system parameter vector. Each Kalman Filter is accompanied by a deterministic controller which is designed based on the same parameter vector. The filters' *a posteriori* probabilities then are used as a measure of confidence in the individual filter's parameter vector and the overall control signal of the MMA-Controller is computed as a weighted sum of the control signals generated by the individual controllers, weighted by the confidence in the respective Kalman Filter.
However, these approaches did not involve adaptation of the individual models (the term "adaptive" rather referred to the change in the weighting factors than to true adaptation of parameters) and thus more or less relied on the number of possible parameter values being *finite*. Furthermore, as pointed out e.g. in [ACKP+77], no stability results were obtained. Finally, in contrast to the approach to be outlined later in this chapter, no switching but weighting was performed.

In a context of adaptive identification and control, the pioneering contribution towards switching between multiple models probably has to be awarded to Martensson [Mar86]. In what followed, so-called *direct* switching schemes were the first to evolve. In those approaches, see e.g. [FB86], the sequence of controllers to switch to is predefined and the time instants at which the switching occurs depend on the output of the system. In contrast to that, in the *indirect* switching schemes, the performance of the multiple models is used to decide when and to which model to switch.[i] The idea was first

[i] The terminology of direct and indirect switching is a bit unfortunate because prone to lead to confusion with direct and indirect adaptive control. Whereas the switching scheme decides when to use which controller, regardless of the structure of the individual controllers, the indirect or direct

presented by Middleton et. al. [MGHM92] and subsequently formalized and extended by Narendra and Balakrishnan in [NBC95, NB97]. In schemes of this kind, multiple estimators for the plant parameters are set up and, similar to the earlier approaches involving Kalman Filters, one indirect adaptive controller is computed from each of the estimators' parameter vectors. The estimators are run in parallel, the "best" (according to some performance criterion which is typically based on the identification errors) is chosen at every instant of time and the according indirect adaptive controller is used to control the plant.

Here it also becomes apparent, why the adaptive control approach adopted has to be the indirect one: The switching has to be based on some kind of performance criterion, but if the direct approach was to be used, only the performance of the active controller could be evaluated and this just after application of the control input generated by this controller. That is, in the case of the direct approach, there is no way of making the decision, which controller is the best one to control the plant at the given instant of time *a priori*. In contrast to that, adaptive estimation of plant parameters like described e.g. in Section 3.1 does not interfere with the operation of the plant, is a passive procedure and thus arbitrarily many estimators can be run at the same time and their performance can be assessed by their individual identification errors. The identification error of the associated estimator in turn can be used as a measure of confidence in the individual controllers and switching schemes can be based on that. It should be stressed here that this decision is purely heuristic because there is no theoretical result showing that a parameter error with a smaller norm ensures a smaller identification error and neither of those guarantee faster convergence of identification or control errors.

4.2.2 Multiple Models, Switching and Tuning

Representatively for the "classical" multiple model approach, we shall now describe the multiple model scheme proposed in [NX00][ii] in some detail and also give a proof of stability. As pointed out earlier, adaptive control using a single model is sufficient as long as the initial parameter error $\|\tilde{\vartheta}(0)\|$ can be kept small and the parameters vary only slowly with time (or not at all). If at least one of the two assumptions is not fulfilled, which is the case e.g. in the presence of large external disturbances, if the knowledge of the parameter values is very limited or in the case of failures in system components, then the case can be made for multiple model approaches.

The particular one that is going to be described here is made of three main ingredients: A bank of identification models, a bank of controllers based on those identification models and a switching rule which evaluates the performance of the individual identifiers and

with regard to adaptive control refers to which parameters are dynamically adjusted: The ones of the controller as in direct adaptive control or the ones of an estimator from whose parameter vector the control parameters are then computed via an algebraic relationship.

[ii] The paper is a summary of the much more detailed analysis in the PhD-Thesis [Xia00], from which most of the proofs in this chapter are taken.

switches to the controller associated with the identifier with the best performance. A general architecture is shown in Figure 4.4.

Three principal structures regarding the individual models can be found in the literature:

1. All models are fixed
2. All models are adaptive
3. Some models are fixed and some are adaptive

The case of all fixed models is quite similar to the early approaches like [Lai76] etc., discussed in the last section. The advantage of fixed models over adaptive models obviously is the computational inexpensiveness. On the other hand, we can say that for the performance to be satisfactory (or even stable), at least one model must be close in some sense to the real plant. Since the plant is assumed to be unknown, this in turn means that the space of possible parameter values must be covered densely enough with fixed models which can require - depending on how big that space of possible parameter values is - a very large number of fixed models.

If all models are adaptive, stability is not an issue, since one adaptive model already assures that. The computational complexity however increases substantially in relation to all fixed models. Another disadvantage shows, if the approach is used to deal with time-varying parameters: Assume, that the parameters stay constant for a while. If the input is persistently exciting this will cause all the models to converge to the same point in parameter space and invalidate the approach. If the input is not PE all models still are going to converge to some subspace of the parameter space, so any strategic distribution of models over parameter space is bound to be destroyed after some running time.

That makes the third structure the method of choice since it combines the advantages of the two others: Computational inexpensiveness and fast response from the fixed models, stability and good asymptotic performance from the adaptive models. Following this considerations, it was suggested in [NB97] to set up M models, $M-2$ of which are fixed and distributed over parameter space. The remaining two models are adaptive, one is free-running and the other is resettable, that means whenever one of the fixed models is chosen because it has the best performance index, this adaptive model initiates adaptation with the parameter vector of the fixed model as its initial value.

Formal description. Let the plant model to be identified be given by the usual d-step ahead predictor form

$$y(k) = \vartheta^{*T}\varphi(k-d), \tag{4.2}$$

where φ and ϑ^* are defined as usual, e.g. in Equation (2.16). The assumptions that apply are exactly the same as in Lemma 3.6, they are repeated here for convenience:

Assumptions 4.1.

(i) The delay d is known exactly.

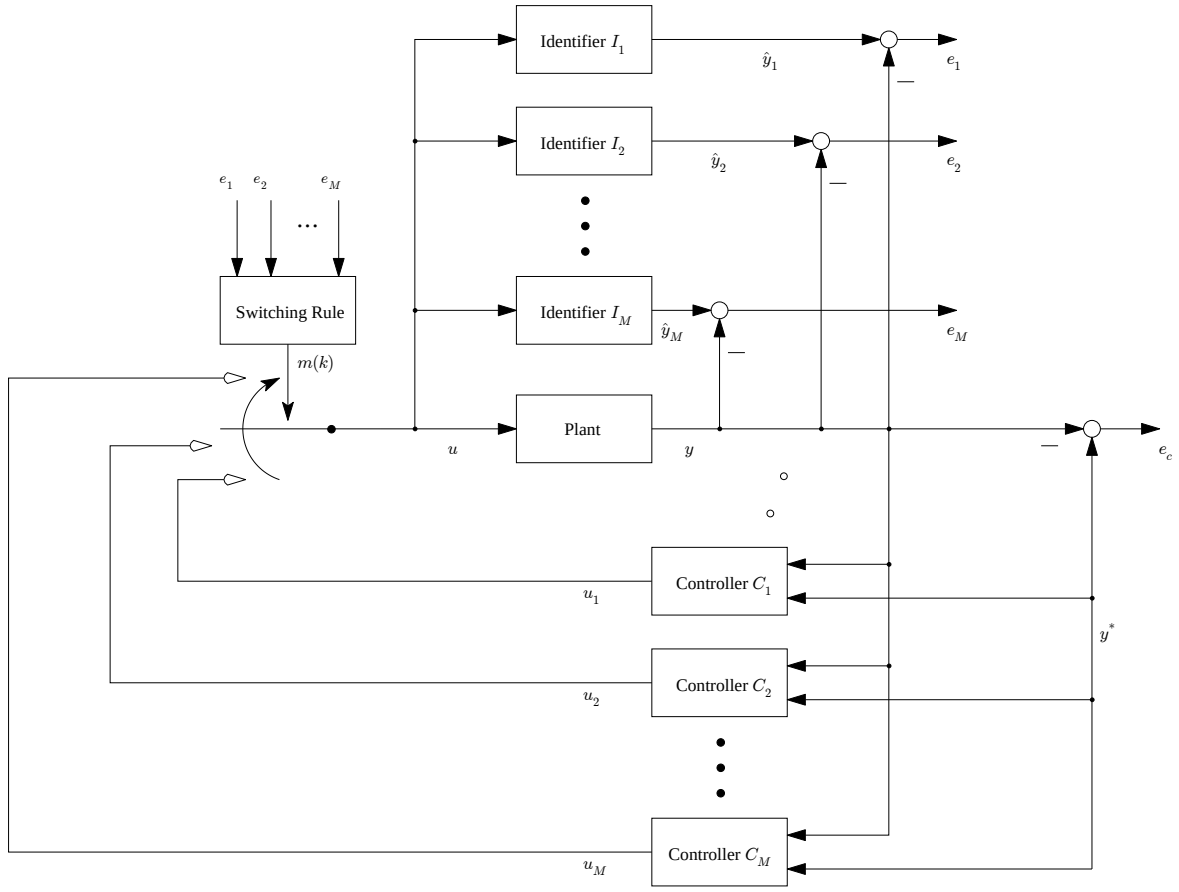

Figure 4.4: An architecture of the multiple model adaptive control approach

(ii) The sign of b_{d+n-1} and a lower bound $\underline{b}$ for $|b_{d+n-1}|$ are known.

(iii) The order n, or at least an upper bound $\overline{n} \geq n$ for the order n, is known.

(iv) The inverse system $B(q)z(k) = A(q)v(k)$ is BIBO stable.

The M identification models are then set up as

$$\hat{y}_i(k) = \vartheta_i^T \varphi(k-d) \qquad i = 1, \ldots, M-2 \tag{4.3a}$$

for the fixed models,

$$\hat{y}_a(k) = \vartheta_a^T(k-1)\varphi(k-d) \tag{4.3b}$$

for the free running adaptive model, and

$$\hat{y}_r(k) = \vartheta_r^T(k-1)\varphi(k-d) \tag{4.3c}$$

for the resettable adaptive model. For each model, one associated one-step-ahead controller is set up which computes its output $u_i(k)$ such that

$$\begin{aligned} y^*(k+d) &= \vartheta_i^T \varphi(k) \qquad i = 1, \ldots, M-2 \\ y^*(k+d) &= \vartheta_i^T(k)\varphi(k) \qquad i = a, r. \end{aligned} \tag{4.4}$$

It now remains to be decided, which of the M possible inputs should be used to control the plant at each instant of time k. As described earlier we hereby want to use the output of the controller in whose parameter vector we have the most confidence. Therefore each of the identifiers gets assigned a performance criterion which incorporates the normalized identification errors from the initial time k_0 until the current time k, and an optional forgetting factor $\rho \in (0, 1]$ that helps putting more weight on recent data

$$J_i(k) = \sum_{\tau=k_0}^{k} \rho^{k-\tau} \left(\frac{\hat{y}_i(\tau) - y(\tau)}{\sqrt{c + \varphi^T \varphi(\tau - d)}} \right)^2 = \sum_{\tau=k_0}^{k} \rho^{k-\tau} \frac{e_i^2(\tau)}{c + \varphi^T \varphi(\tau - d)}, \tag{4.5}$$

where $c > 0$ is some constant and $i = 1, \ldots, M-2, a, r$, and we choose

$$\begin{aligned} u(k) &= u_{m(k)}(k) \\ m(k) &= \arg \min_{i=1,\ldots,M-2,a,r} J_i(k). \end{aligned} \tag{4.6}$$

As the adaptive law which governs the parameter updates $\vartheta_a(k) - \vartheta_a(k-1)$ and $\vartheta_r(k) - \vartheta_r(k-1)$, any stable adaptive algorithm can be chosen.
A slight modification is necessary for the resettable model: Whenever one of the fixed models becomes active (i.e. is chosen for the first time in a finite interval) the parameter vector $\vartheta_r(k)$ is reset to the value of the parameter vector of the fixed model, so if for instance the projection algorithm is used:

$$\vartheta_r(k) = \begin{cases} \vartheta_r(k-1) - \gamma \varepsilon_r(k)\varphi(k-d) & \text{if } m(k) \in \{m(k-1), a, r\} \\ \vartheta_{m(k)} & \text{if } m(k) \notin \{m(k-1), a, r\} \end{cases} \tag{4.7}$$

Proof of stability. The proof of stability mainly consists of showing, that the switching stops after a *finite* time T at one of the adaptive models, that by themselves already assure stability. Going along the lines of [Xia00], we first prove stability for $\rho = 1$ and afterwards show, how stability for $\rho \in (0, 1)$ follows from that with an auxiliary lemma.

Lemma 4.2 (Stability of multiple model adaptive control, $\rho = 1$)**.** *If the same assumptions on the plant as in Lemma 3.6 can be made and a stable identification scheme is used along with the control law (4.4), the performance criterion (4.5), $\rho = 1$ and the switching scheme (4.6), then $\vartheta_a(k), \vartheta_r(k), u(k) \in \ell^\infty$ and $e_c(k) = y(k) - y^*(k) \to 0$.*

Proof. That $\vartheta_a(k), \vartheta_r(k) \in \ell^\infty$ follows directly from Lemma 3.2, since no changes to the identification schemes are made. So we only have to show that $u(k) \in \ell^\infty$ and $e_c(k) \to 0$. Since the performance indices $J_i(k)$ are monotonic, we have that $\lim_{k\to\infty} J_i(k)$ either is a finite constant K_i or ∞. For all fixed models, the latter is (almost always) the case. To see that, observe that from $\|\varphi(k-d)\| \leq D_1 + D_2 \max_{\tau\leq k} |e_i(\tau)|$ (from the proof of Lemma 3.6) it follows, that $\frac{e_i^2(k)}{1+\varphi^T\varphi(k-d)} \geq \frac{e_i^2(k)}{D_1^2+D_2^2 \max_{\tau\leq k} e_i^2(\tau)}$ which only tends to zero if $e_i \to 0$. From $e_i = \tilde{\vartheta}_i^T \varphi(k-d)$ it then follows that this can be the case if either $\varphi(k-d) \to 0$, which is a trivial case, or $\tilde{\vartheta}_i \equiv 0$, which would mean that one of the fixed models is an exact plant model, or in the singular case that $\tilde{\vartheta}_i \perp \varphi(k-d)$ from some time k_i on. So except for pathological cases, $J_i \to \infty$ for all fixed models, from which it follows that after a finite time T, $J_a(k) < J_i(k)$ for all $i = 1, \ldots, M-2$ and that means that after a finite time, no fixed models will be selected anymore and since the adaptive model alone already ensure stability, we have $e_c \to 0$.
Note that the resettable model will never be reset after T, since no fixed models are being selected, and by that does not change the stability properties. □

To show how that extends to $\rho \in (0, 1)$, the following auxiliary lemma is needed:

Lemma 4.3 ([Xia00])**.** *Let $a, b : \mathbb{Z}^+ \to \mathbb{R}^+$ with*

$$\sum_{\tau=1}^{\infty} a(\tau) < \infty$$

and

$$\sum_{\tau=1}^{\infty} b(\tau) = \infty.$$

If there exists a lower bound $\delta > 0$ such that $\max_{j\in\mathbb{Z}^+}\{n \,|\, b(j) < \delta, b(j+1) < \delta, \ldots, b(j+n) < \delta\} = L < \infty$ (i.e. if the number of consecutive elements of b that are smaller than δ is at most $L+1$), then there exists a finite T, such that for all $k \geq T$:

$$\sum_{\tau=1}^{k} \rho^{k-\tau} b(\tau) > \sum_{\tau=1}^{k} \rho^{k-\tau} a(\tau).$$

Proof. From the assumptions on b it follows, that for $k > L$ there exists a j such that

$$\sum_{\tau=1}^{k} \rho^{k-\tau} b(\tau) \quad \geq \quad \sum_{\tau \in \{1,\ldots,k\} \setminus \{j,\ldots,j+L\}} \rho^{k-\tau} \delta \quad + \quad \sum_{\tau \in \{j,\ldots,j+L\}} \rho^{k-\tau} b(\tau) \quad \geq \quad \rho^{L+1} \delta.$$

On the other hand, since $a \in \ell^1$, $a \to 0$, so for any $\varepsilon_1 > 0$ there exists $N_{\varepsilon_1} > 0$ such that $a(k) < \varepsilon_1 \; \forall \, k \geq N_{\varepsilon_1}$, or

$$\sum_{\tau=1}^{k} \rho^{k-\tau} a(\tau) \leq \sum_{\tau=1}^{N_{\varepsilon_1}} \rho^{k-\tau} a(\tau) + \sum_{\tau=N_{\varepsilon_1}}^{k} \rho^{k-\tau} \varepsilon_1 \leq \sum_{\tau=1}^{N_{\varepsilon_1}} \rho^{k-\tau} a(\tau) + \frac{\varepsilon_1}{1-\rho}.$$

Since $a \in \ell^\infty$, the first sum tends to zero as $k \to \infty$, so $\forall \varepsilon_2 > 0 \; \exists N_{\varepsilon_2} > 0$: $k \geq N_{\varepsilon_2}$ $\Rightarrow \sum_{\tau=1}^{N_{\varepsilon_1}} \rho^{k-\tau} a(\tau) < \varepsilon_2$. Now choose ε_1 and ε_2 such that $\frac{\varepsilon_1}{1-\rho} + \varepsilon_2 < \rho^{L+1}\delta$. The result follows with $T = \max\{N_{\varepsilon_2}, N_{\varepsilon_1}\}$. □

Corollary 4.4. *Lemma 4.2 also holds for $\rho \in (0, 1)$.*

Proof. Application of Lemma 4.3 with $b(k) = \frac{e_i^2(k)}{1+\varphi^T \varphi(k-d)}$ and $a(k) = \frac{e_a^2(k)}{1+\varphi^T \varphi(k-d)}$ to the proof of Lemma 4.2 extends it to the case $\rho \in (0, 1)$. □

The stability results reveal, that the multiple model approach inherits its stability from the single model adaptive control whereas the expected performance improvement remains based on merely heuristic considerations. The improvement of the performance and thus the applicability of multiple model approaches has been shown by extensive simulation studies in the literature that has been mentioned in this chapter. We will conclude our discussion by showing that the approach does also improve performance in the introductory examples.

4.2.3 Simulation: Multiple-Model Adaptive Control of a system with discontinuous time-variance

We now go back to the introductory examples and show, how the approach of this section is apt to improve performance in this cases considerably. For this simulations, the same conditions as in Section 4.1 are used. The controller consists of four fixed models with parameter estimates $\vartheta_i = \vartheta_i^* \bullet \begin{bmatrix} 1.1 & 1.05 & 0.9 & 0.95 & 0.9 & 1.1 \end{bmatrix}^T$, where ϑ_i^* corresponds to the d-step ahead predictor form of the SPM $y(k) = \bar{\varphi}^T(k)\bar{\vartheta}_i^*$, that is relatively close to the four possible parameter vectors of the plant. The adaptive models are initialized with the "bad" initial estimates from Section 4.1, that is $\vartheta_r(3) = \vartheta_a(3) = \vartheta_1^* \bullet \begin{bmatrix} 1.3 & 1.3 & 0.8 & 0.7 & 0.8 & 1.3 \end{bmatrix}^T$ and the forgetting factor is chosen to be $\rho = 0.35$, a good value to deal with the time-variations.

Figure 4.5 shows only the first 100 instants of time. Compared with Figure 4.2 it is obvious that the transients decreased in magnitude and decay a lot faster. The reason is

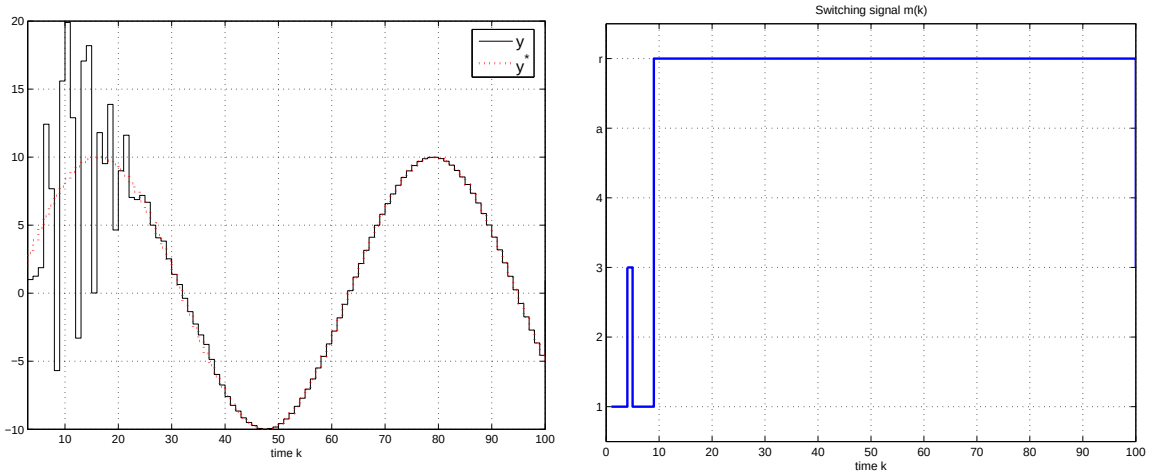

Figure 4.5: The multiple model approach improves the performance in the transient phase considerably.

that the switching scheme identifies the closest model quickly and adaptation is initiated from there.

In Figure 4.6, 600 instants of time are shown. In comparison with Figure 4.3 it can be seen that although spikes can be observed, they are several orders of magnitude smaller than in the single model case, where the response even became unstable after a while. The right plot shows that the performance criterion always finds the best estimate of the parameter vector among the fixed models after a very short period of time (and with a little confusion every now and then). Adaptation is initiated from there and the resettable adaptive model converges faster than the free-running one, which leads to the resettable model being active for most of the time.

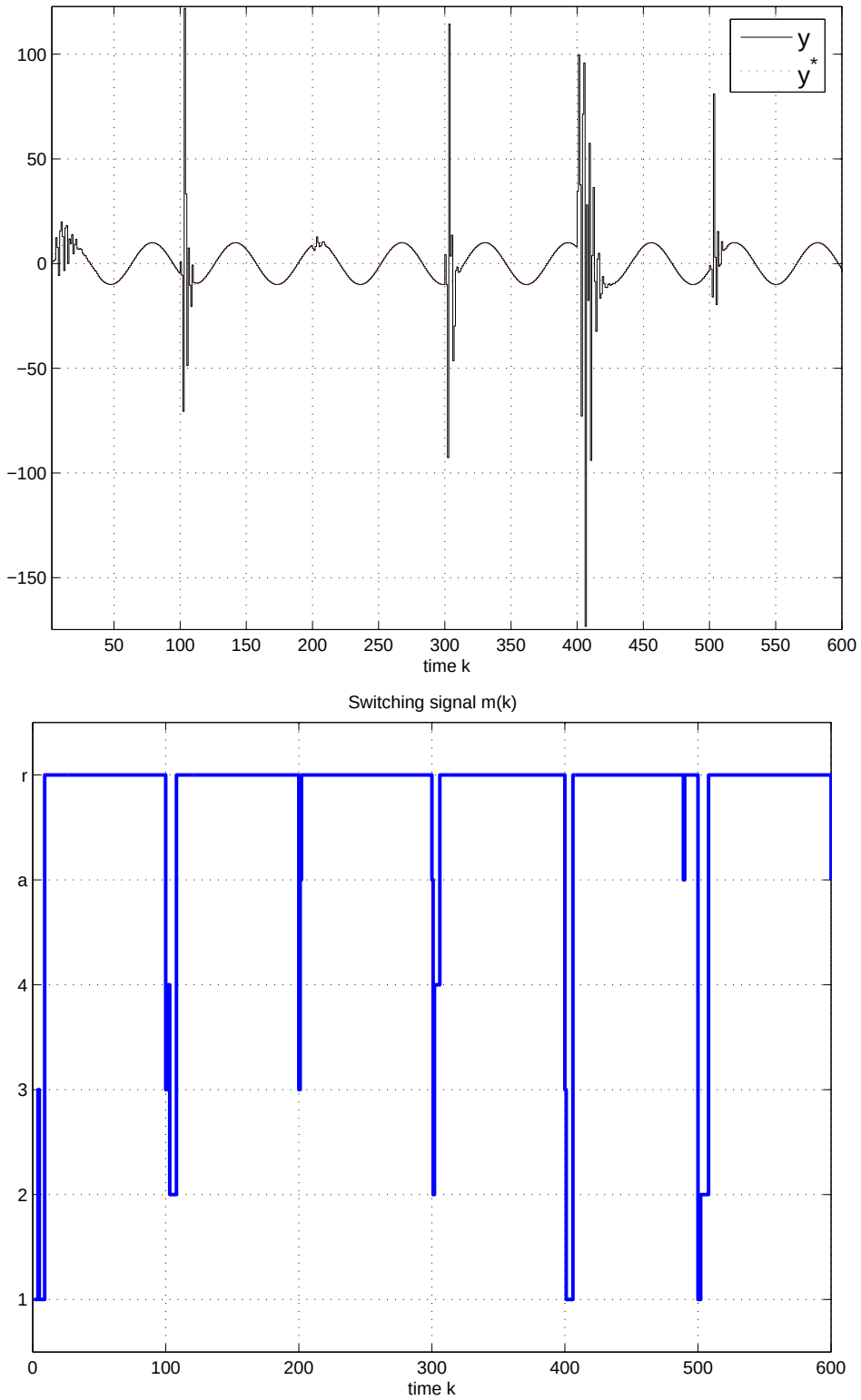

Figure 4.6: The performance in the presence of rapid parameter changes improved considerably, instability cannot be observed anymore.

5 A new method of Identification and Control of time-varying linear plants

The limitations of the methods we have presented so far are readily summarized: Time-variations and transients. In other words, any variation of the parameters leads to a transient in the response. If the variations are happening too fast, so that the identification algorithm jumps from transient to transient without time to settle in between, instability results. In this chapter we will present a new method to handle time variations of (almost) arbitrary speed in a plant that switches between a finite number of subsystems. The method, which has been proposed in [FN06], involves a Multiple-Model controller with adaptive models of the subsystems that can asymptotically provide tracking d time-steps after every switch, provided that switching does not occur on intervals with a minimum length of at least d.

Under the assumption that it is possible to identify all the subsystems exactly, it is easily shown how a very simple controller yields the desired response. So the next step is to provide an algorithm which ensures exactly that: Convergence of multiple models to multiple subsystems, regardless of the speed of switching.

5.1 The problem formulation

The plant. To make the problem of a time-variant plant tractable, we have to assign a certain structure to the time-variations. This structure shall be the easiest one can think of: The plant is assumed to switch between a finite number of subsystems:

$$y(k) = \varphi^T(k-d)\vartheta^*(k-d), \tag{5.1}$$

where $\vartheta^* : \mathbb{Z}^+ \to \Theta^* \subset \mathbb{R}^p$ and $\Theta^* = \{\vartheta_1^*, \ldots, \vartheta_\nu^*\}$. We shall refer to the time-invariant systems described by $y(k) = \varphi^T(k-d)\vartheta_i^*$, $i \in N$, as the *subsystems* of the plant, $p \in \mathbb{Z}^+$ is the dimension of the parameter vector and the index set $N = \{1, \ldots, \nu\}$ is introduced for ease of notation.

Furthermore, the following technical assumptions have to be made:

Assumptions 5.1.

(i) The switching never stops, that is for every $k \geq 1$ there exist $k_i \geq k$, $i \in N$, such that $\vartheta^*(k_i) = \vartheta_i^*$.

(ii) The input delay d is the same for all subsystems.

(iii) The plant behaviour stays constant for at least $d+1$ instants of time, that is if the switching instants are denoted by s_i, then $s_{i+1} - s_i \geq d+1 \ \forall i \in \mathbb{Z}^+$.

(iv) Assumptions 4.1 are fulfilled for every subsystem to make 1-step-ahead control possible.

The control task. The standard goal of asymptotic tracking has to be somewhat redefined for plants of the form just described. A certain inherent control error cannot be avoided, even with an exact model for every subsystem, because every change in the plant behaviour will stir up the control error, so in general convergence to zero cannot be achieved and we will have to settle for convergence to zero on an unbounded *subsequence* of $\mathbb{Z}^+$. We leave the detailed analysis which will reveal the exact cause for trouble and the nature of the control error convergence to Section 5.3, where the control algorithm is described and analysed.

5.2 A short overview of the notation

Before we proceed to describe the control and identification algorithms it seems advisable to recapitulate the notation introduced so far, since it could be a source for confusion otherwise.

- ▷ The parameter vector of the plant ϑ^* is now replaced by a piecewise constant function $\vartheta^*(k)$. The range of this function is the finite set Θ^*, which consists of the constant parameter vectors ϑ_i^* of the ν subsystems of the plant.
- ▷ Instead of fixed models we now have only adaptive models, each with its own parameter estimate $\vartheta_i(k)$. One model is chosen at each instant of time to be the one with the best performance. Its index is denoted by $m(k)$. The vector $\vartheta(k) = \vartheta_{m(k)}(k)$ is the parameter estimate used for control and thus in some sense the "actual" parameter estimate.
- ▷ The signal $y^*(k)$ is the reference signal.
- ▷ Each model has the output $\hat{y}(k) = \vartheta_i^T(k-1)\varphi(k-d)$ and an identification error $e_i(k) = \hat{y}(k) - y(k)$. Opposed to that there is only *one* control error $e_c(k) = y(k) - y^*(k)$.

5.3 The control algorithm

Assume for the moment, that all subsystems have been identified exactly, that is for any $\vartheta_i^* \in \Theta^*$ there is a model j such that $\vartheta_j = \vartheta_i^*$. Since we now explicitly assume a time-variant system, it does not make much sense to use a stability criterion with memory like (4.5) anymore. Instead, only the instantaneous identification error is taken into account, that is

$$J_i(k) = e_i^2(k) = (\hat{y}(k) - y(k))^{2\downarrow i\downarrow}. \tag{5.2}$$

Like in the classical multiple model approach of Section 4.2, for each model a corresponding 1-step-ahead controller is set up. The outputs of the controllers are denoted by u_i and at each instant of time the input u is chosen as

$$\begin{aligned} u(k) &= u_{m(k)}(k) \\ m(k) &= \arg\min_{i \in N} J_i(k), \end{aligned} \tag{5.3}$$

in other words, the estimate of the parameter vector is $\vartheta(k) = \vartheta_{m(k)}$. Since we assumed exact estimates, we have that $J_i(k) = \varphi^T(k-d)\,(\vartheta_i^* - \vartheta^*(k-d)) = 0$ for one $i \in N$, and thus $m(k) = i$, which leads to

$$\vartheta(k) = \vartheta_{m(k)}^* = \vartheta^*(k-d) \qquad\qquad \forall k \in \mathbb{Z}^+, \tag{5.4}$$

so a change in the parameter vector can only be detected d instants of time later, which is quite intuitive.

Now let s be a time instant at which the plant switches from subsystem j to subsystem i, $i \neq j$, so regarding Assumptions 5.1 that means that $\vartheta^*(k) = \vartheta_j^*$ at least for $k \in [s-d-1, s)$ and $\vartheta^*(k) = \vartheta_i^*$ for $k \in [s, s+d]$. Expanding the control error $e_c(k) = y(k) - y^*(k)$ to

$$\begin{aligned} e_c(k) &= \varphi^T(k-d)\left[\vartheta^*(k-d) - \vartheta_{m(k-d)}^*\right] \\ &= \varphi^T(k-d)\left[\vartheta^*(k-d) - \vartheta_{m(k)}^* + \vartheta_{m(k)}^* - \vartheta_{m(k-d)}^*\right] \\ &= \varphi^T(k-d)\underbrace{\left[\vartheta^*(k-d) - \vartheta_{m(k)}^*\right]}_{\equiv 0,\ \text{cf. (5.4)}} + \varphi^T(k-d)\left[\vartheta_{m(k)}^* - \vartheta_{m(k-d)}^*\right] \\ &= \varphi^T(k-d)\left[\vartheta_{m(k)}^* - \vartheta_{m(k-d)}^*\right]. \end{aligned} \tag{5.5}$$

So from Equation (5.4) it follows, that

$$e_c(k) \neq 0 \qquad \text{for } k \in I_c(s) := \{s+d, s+d+1, \ldots, s+2d-1\}, \tag{5.6}$$

because for $k \in I_c(s)$, $m(k) = i$ and $m(k-d) = j$. $I_c(s)$ is referred to as the *inherent control error interval*, because for all switching instants s, the control error during that

$^{\downarrow i\downarrow}$With the convention $0^0 = 1$, this could be seen as the performance criterion from (4.5) with $\rho = 0$ and without the normalization.

interval could only be avoided by switching from model j to model i at time $k = s$, that is d instances before the change can be detected at the output. See also Figure 5.1 for an illustration.

If we now substitute the assumption that we have exact estimates for all subsystems with the weaker assumption that there exists a stable identification scheme which simultaneously identifies all subsystems, we can state the following theorem.

Theorem 5.2 ([FN06]). *Suppose we have a plant as described in Section 5.1, subject to Assumptions 5.1, and an adaptive algorithm which ensures that for any $i \in N$ there is a $j \in N$ such that $\|\vartheta_j(k) - \vartheta_i^*\| \to 0$ as $k \to \infty$. Then the controller*

$$\begin{aligned} u(k) &= u_{m(k)}(k) \\ m(k) &= \arg\min_{i \in N} J_i(k) \\ u_i(k) \text{ such that } y^*(k+d) &= \varphi^T(k)\vartheta_i(k-1) \end{aligned} \tag{5.7}$$

stabilizes the plant in the sense that $\|\varphi(k)\| \in \ell^\infty$ and $e_c(k) \to 0$ on a subsequence $K_l \subset \mathbb{Z}^+$.

Proof. For simplicity and without loss of generality let the identification models be rearranged such that $\|\vartheta_i(k) - \vartheta_i^*\| \to 0$. Then it follows that after some time $\bar{k} \in \mathbb{Z}^+$ the identification error $|e_i| \leq \|\vartheta_i(k-1) - \vartheta^*(k-d)\| \, \|\varphi(k-d)\|$ is smaller than the identification errors $|e_j(k)|$, $i \neq j$, of all the other models whenever $\vartheta^*(k-d) = \vartheta_i^*$, or in other words, that $J_i(k) < J_j(k)$ $\forall j \neq i$ $\forall k \in \{k \,|\, \vartheta^*(k-d) = \vartheta_i^*\}$.

That means we can define an unbounded subsequence $K_n = \mathbb{Z}^+ \setminus Q_1 \subset \mathbb{Z}^+$, where $Q_1 = \{s+d \,|\, s \in \mathcal{S}\}$ ($\mathcal{S}$ denotes the set of switching instants), such that $m(k-1) = m(k)$ and $e_{m(k)} = (\vartheta(k-1) - \vartheta^*(k-d))^T \varphi(k-d) \to 0$ on this subsequence, because $m(k) = i$ if $\vartheta^*(k-d) = \vartheta_i^*$ and $k \in K_n$.

We can define another unbounded subsequence $K_m = \mathbb{Z}^+ \setminus Q_2 \subset \mathbb{Z}^+$, where $Q_2 = \{s+d+1, \ldots, s+2d-1 \,|\, s \in \mathcal{S}\}$, such that $\|\vartheta(k-1) - \vartheta(k-d)\| \to 0$ on this subsequence, because $m(k-d) = m(k-1)$ and thus $\vartheta(k-1)$ and $\vartheta(k-d)$ belong to the same model for $k \in K_m$.

So the set Q_1 includes all the "detection" instants, and $Q_1 \cup Q_2 = I_c$, the inherent control error interval. The construction of Q_1 and Q_2 is illustrated in Figure 5.1.

Like in the proof of Lemma 3.6 we can derive the relationship

$$\frac{|e_c(k)|}{\sqrt{c + \varphi^T\varphi(k-d)}} \leq \frac{|e_{m(k-1)}(k)|}{\sqrt{c + \varphi^T\varphi(k-d)}} + \frac{\|\varphi(k-d)\| \, \|\vartheta(k-1) - \vartheta(k-d)\|}{\sqrt{c + \varphi^T\varphi(k-d)}}, \tag{5.8}$$

where $e_{m(k-1)}(k) = \hat{y}_{m(k-1)}(k) - y(k)$ is the *a priori* identification error, that is the identification error at time k before the new active controller is chosen according to (5.7). If we restrict our attention to the subsequence $K_l = K_m \cap K_n = \mathbb{Z}^+ \setminus I_c$ it follows

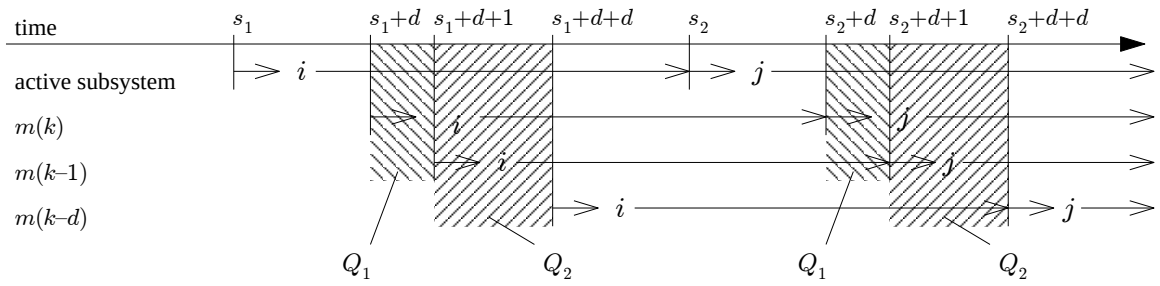

Figure 5.1: An illustration of the sets Q_1 and Q_2 in the proof of Theorem 5.2.

that $\frac{|e_c(k)|}{\sqrt{c+\varphi^T\varphi(k-d)}} \to 0$ on K_l. Since we assumed that $s_i - s_j \geq d+1$ it follows that each interval of length greater than $d+1$ (after $\bar{k}$ has passed) contains at least one element of K_l, and thus K_l is unbounded.

In a manner similar to the proof of Lemma 3.6 it can be shown now that from that result, namely that the normalized control error $\frac{|e_c(k)|}{\sqrt{c+\varphi^T\varphi(k-d)}}$ tends to zero on K_l, it follows that also the control error e_c tends to zero on K_l:
Since we assumed a stable inverse for all possible subsystems, it follows that the input and thus the regression vector of the system cannot grow faster than the control error. Following the exact same steps this can be used to show that, if the time axis for all sequences is taken to be K_l instead of $\mathbb{Z}^+$

- if $e_c \in \ell^\infty$, then $\|\varphi\| \in \ell^\infty$
- if $\|\varphi\| \in \ell^\infty$, then $e_c \to 0$
- if $e_c \notin \ell^\infty$, then $\frac{|e_c(k)|}{\sqrt{c+\varphi^T\varphi(k-d)}} \not\to 0$, a contradiction,

and so it follows, that $e_c \to 0$ on K_l and $\|\varphi\| \in \ell^\infty$. □

5.4 The identification algorithm

The more interesting question is of course, what an identification algorithm that ensures that ν adaptive models are converging simultaneously to ν different subsystems could look like. First of all, it is important to understand the difference of this approach to the one outlined in Chapter 4.

The now classical Multiple-Model approach was designed to deal with time-invariant plants, consequently theoretic results are strictly valid only in this case. Application to time-varying cases as demonstrated in the last chapter is based on the heuristic assumption that convergence takes place and after a change in parameters a new adaptation problem has to be solved, possibly with bad initial estimates because they now depend

on the active parameter vector before the switch and cannot be chosen close to some expected values of the new parameter vector.[ii]
Since every rapid change in parameters is seen as a new problem, information about the "old" parameter vector is discarded. For this reason it does not seem wise to incorporate more than one adaptive model (or two - with an additional resettable one) because at every time k, all efforts are directed towards "hunting down" the active subsystem, the models thereby act completely decoupled from each other. Thus if the parameter vector remains unchanged for long enough, all adaptive models will converge to the same point in parameter space.

In view of the assumptions in Theorem 5.2, the new identification algorithm must include ν adaptive models, but the effect of all of them converging to a single point must be prevented somehow. It is clear that this calls for interaction of some kind between the models. In [FN06] it has been proposed for this interaction to be a competition amongst the individual models for the maximum adaptive gain.

The detailed algorithm consists of ν adaptive models, each maintaining its own parameter estimate $\vartheta_i(k)$, $i \in N$ and updating it by the adaptive law

$$\vartheta_i(k) = \vartheta_i(k-1) - \eta_i(k)\gamma\varepsilon_i(k)\varphi(k-d). \tag{5.9}$$

The only difference to the projection algorithm (3.7) thereby is the new time-variant factor $\eta_i(k)$ which will introduce the competition between the models. It is straightforward to favour the model that is closest in some sense. Here the same heuristic assumption as in the classical multiple model approach is used: It is assumed that the closest model also has the smallest identification error $e_i(k) = \hat{y}_i(k) - y(k)$[iii] and η_i is computed as

$$\eta_i(k) = \frac{e_i^{-2}(k)}{e_1^{-2}(k) + e_i^{-2}(k) + \cdots + e_\nu^{-2}(k)}. \tag{5.10}$$

Since $\sum_{i\in N} \eta_i(k) = 1$, the gain γ is distributed amongst the models, but the best one gets to move the most, so its advantage over the other models increases. This will lead to an even bigger favouring in the next step and so forth as long as this model stays the best. If one identification error, say $e_j(k)$, tends to zero, we have that $\eta_j \to 1$ and $\eta_i \to 0$, $i \neq j$.

5.4.1 Empiric findings from simulation studies

Since the algorithm has refused to lend itself to a complete theoretical analysis so far, this section is dedicated to some simulations that will support the assumptions that the

[ii] So far this applies to the single model approach, too, the multiple model approach merely speeds up convergence by alleviating the bad-initial-values problem.

[iii] Like in Section 4.2 it should be stressed that this is purely heuristic and that $\|\tilde{\vartheta}_i(k)\| < \|\tilde{\vartheta}_j(k)\| \not\Rightarrow e_i(k) < e_j(k)$ but that the only information about the success of the identification we have is the identification error.

algorithm works and its convergence analysis has to be done along different lines than in the case of classical Adaptive Identification (cf. Lemma 3.2).

The algorithm works

Although no proof of stability could be given so far, numerous simulations give strong evidence that the algorithm works. This is demonstrated with three examples, that are representatives for different circumstances

$p = 2$, $\nu = 5$, **models start close to subsystems.** For this simulation it was assumed that the values of the subsystems are known to some degree, so each of the 5 models can be started close to one of the subsystems. The plant switches between the subsystems in a random sequence and at random instants of time, the input was chosen as $u(k) = 10\sin(k/10)$ which is PE. From the results shown in Figure 5.2 it can be seen that each model is converging to "its" subsystem rather directly. Thus it is no surprise, that the index of the active plant and the index of the maximal $\eta_i(k)$ (that is the index of the model that would be used for control) synchronize quickly, what can be seen in Figure 5.3. Note that here as well as in all the other plots involving the switching signal, the time-axis of the active subsystems is shifted by d instants of time to make the synchronization visible.

$p = 2$, $\nu = 5$, **models start in a cluster.** Now no use is made of prior information about the location of the subsystems. Instead, the models start in a cluster, relatively close to each other and far away from the subsystems. This leads to a lot less smooth and slower convergence, the properties of which shall be discussed later. The point that is supposed to be made here is that the algorithm still leads to convergence, which can be seen in Figure 5.4, and synchronization of the switching instants, cf. Figure 5.5.

$p = 6$, $\nu = 4$, **models start close to subsystems.** This example shows, that the algorithm also works for higher order cases. Here the 2-step ahead predictor form of a plant was identified using the PE input signal $10\sin(k/10) + 15\sin(k/5) + 20\cos(k/2) + 5\cos(k) + 30\cos(2k)$. The time series of the parameter estimates along with their real values are shown in Figure 5.6 and the synchronization of $\max\{\eta_i\}$ with the switching instants of the plant is shown in Figure 5.7.

$p = 6$, $\nu = 4$, **models start in a cluster.** Again, no use of prior information about the subsystem location was made and the models start all close together. Like in the first-order case convergence takes a bit longer and is not as smooth as in the case where each model starts close to one subsystem, but eventually takes place, as can be seen in Figure 5.8.

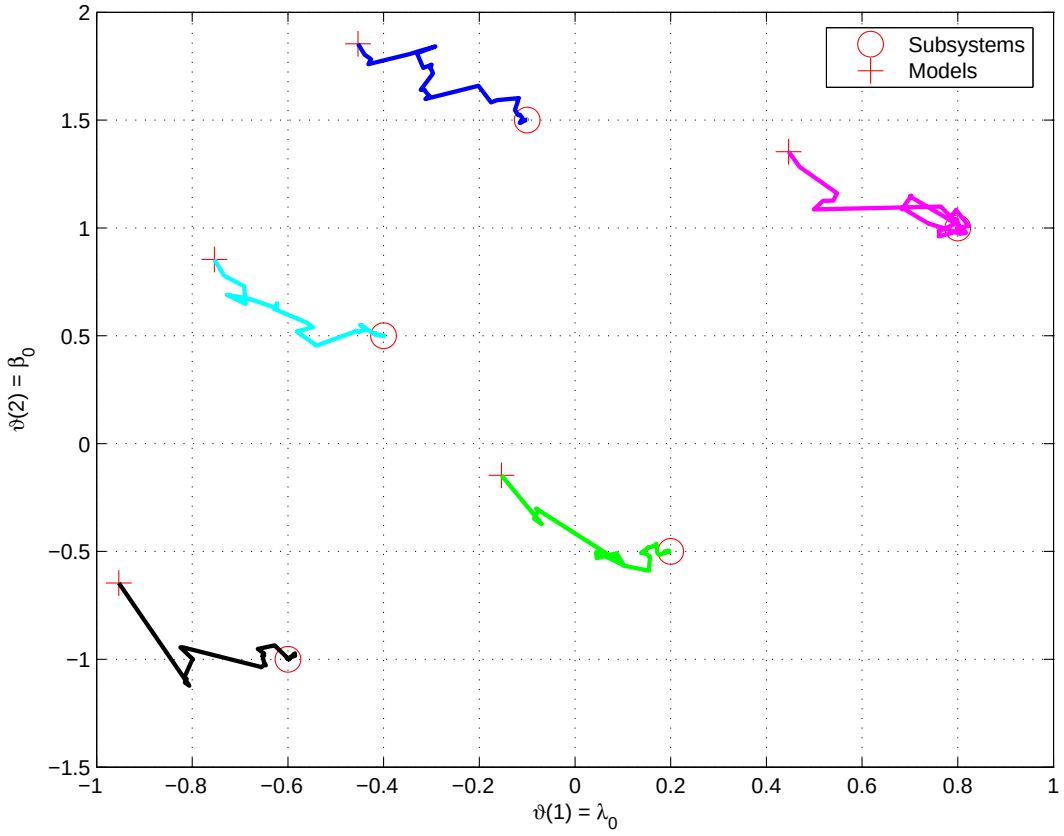

Figure 5.2: Each model starts close to one subsystem and converges to it.

Which model converges to which subsystem is not determined by their locations in parameter space

Foregoing examples and numerous other simulations with different parameter values have shown that in the case where approximate values for the parameters of the subsystems are known and one model is assigned to each subsystem by placing the initial value close to it the intuitive thing happens, namely each model converges to the subsystem closest to it. The intuition that an association of the models with the subsystems is always predetermined by the setup turns out to be misleading though. In fact, examples can be found where the location of models and subsystems is identical and the difference only lies in the model sequence and the switching instances, but the models nevertheless converge to a different subsystem with each run of the simulation as shown in Figure 5.9, which merely consists of 4 reruns of the second example from the preceding section.

Two phases: Self-organization and convergence

In the examples presented so far the system shows a smooth convergence behaviour whenever the models start close to the plants, but a rather wild one whenever the models'

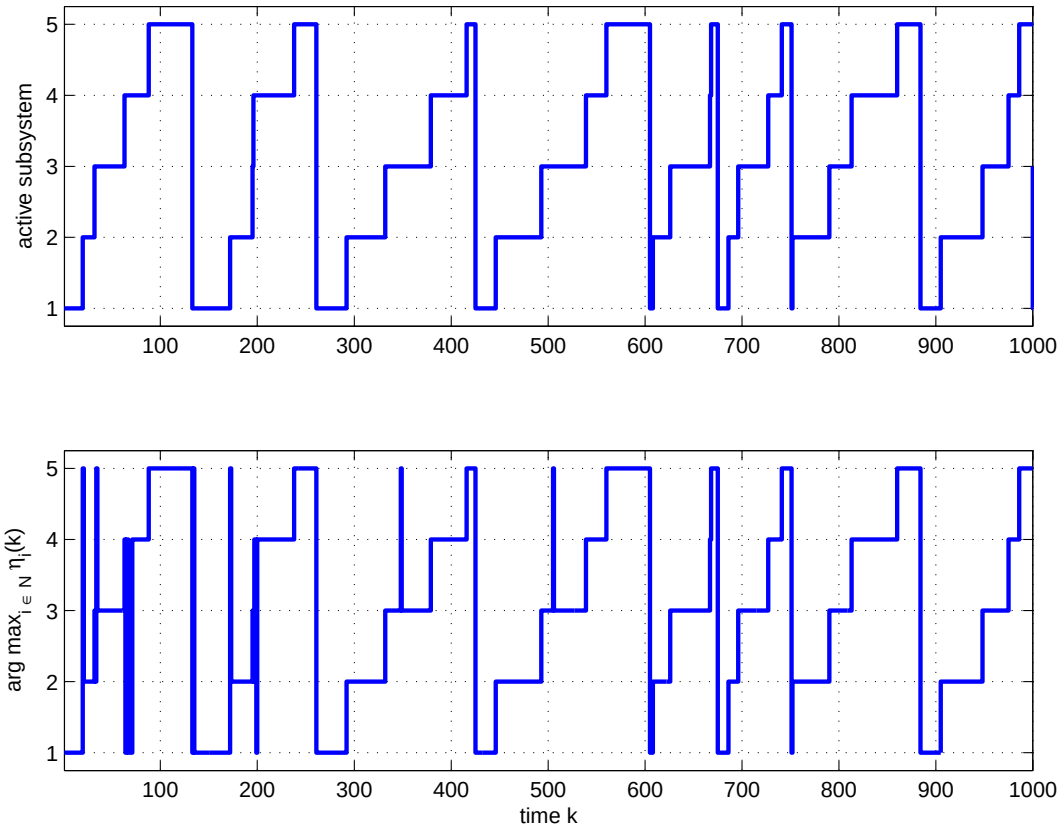

Figure 5.3: Switching of plant and models synchronizes quickly when models start close to subsystems.

initial location is totally unrelated to the subsystems'. One could use this qualitatively different behaviour to distinguish two phases:
In the first one, the models (or rather some of the models) do not seem to really move in the direction of a subsystem but rather wobble around in parameter space until they get separated from each other and by that the decision is made, to which subsystem each model converges. In [FN06], the term *self-organization* was used for this wobbling about until a structure emerges from which convergence can take place.
In the second one, the models converge more or less straight towards "their" subsystems without coming close to each other. One could say that in the case where approximate subsystem locations are known, the system starts in the second phase because it already is organized.

This would be easily illustrated using an animation for $p = 2$, Figure 5.10 shows some snapshots of a typical phase portrait:

- Figure 5.10(a) shows the initial configuration: The models start close to each other in the lower right corner.
- In Figure 5.10(b), model $\square$ seems to already have converged to a subsystem. While

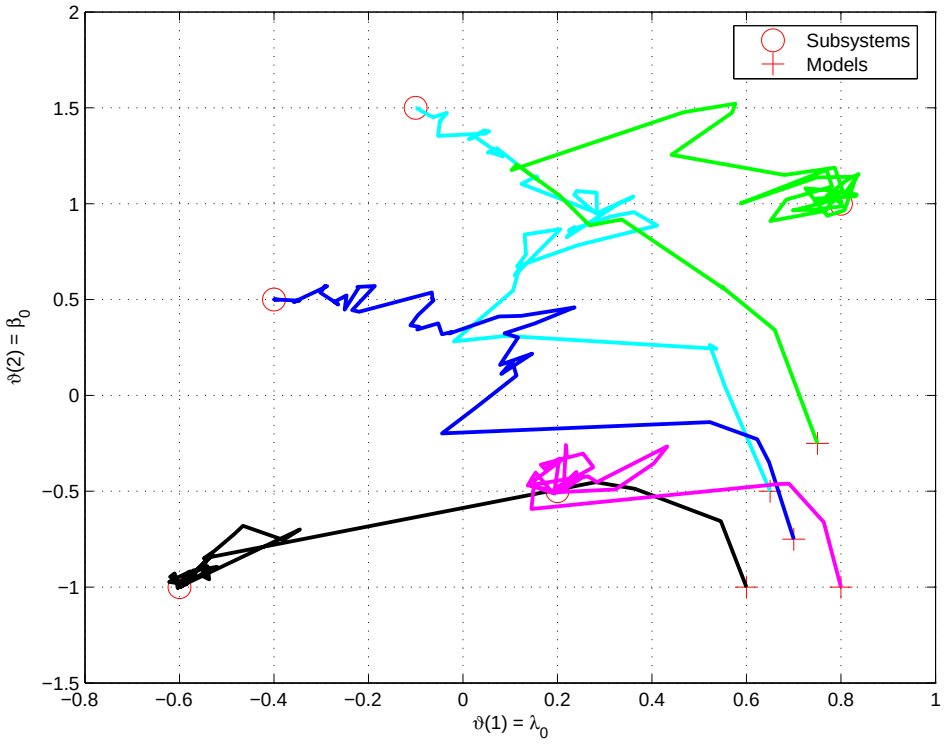

Figure 5.4: Even though the models start with no association to any subsystem convergence takes place.

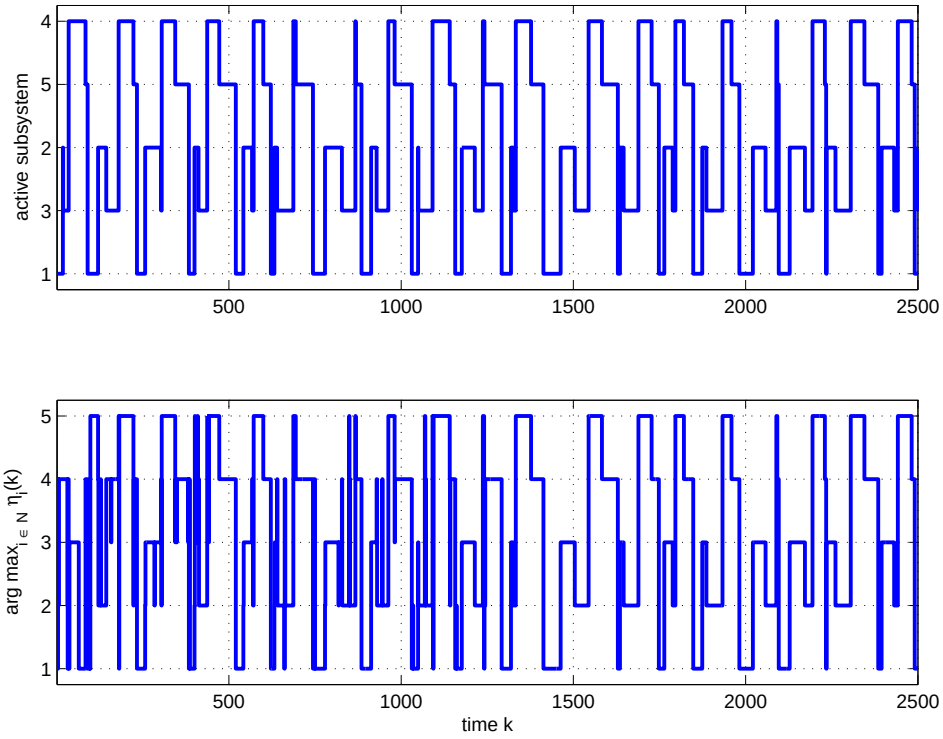

Figure 5.5: Switching of plant and models synchronizes. Note that the subsystems have been rearranged on the ordinate since model i does not necessarily converge to subsystem i.

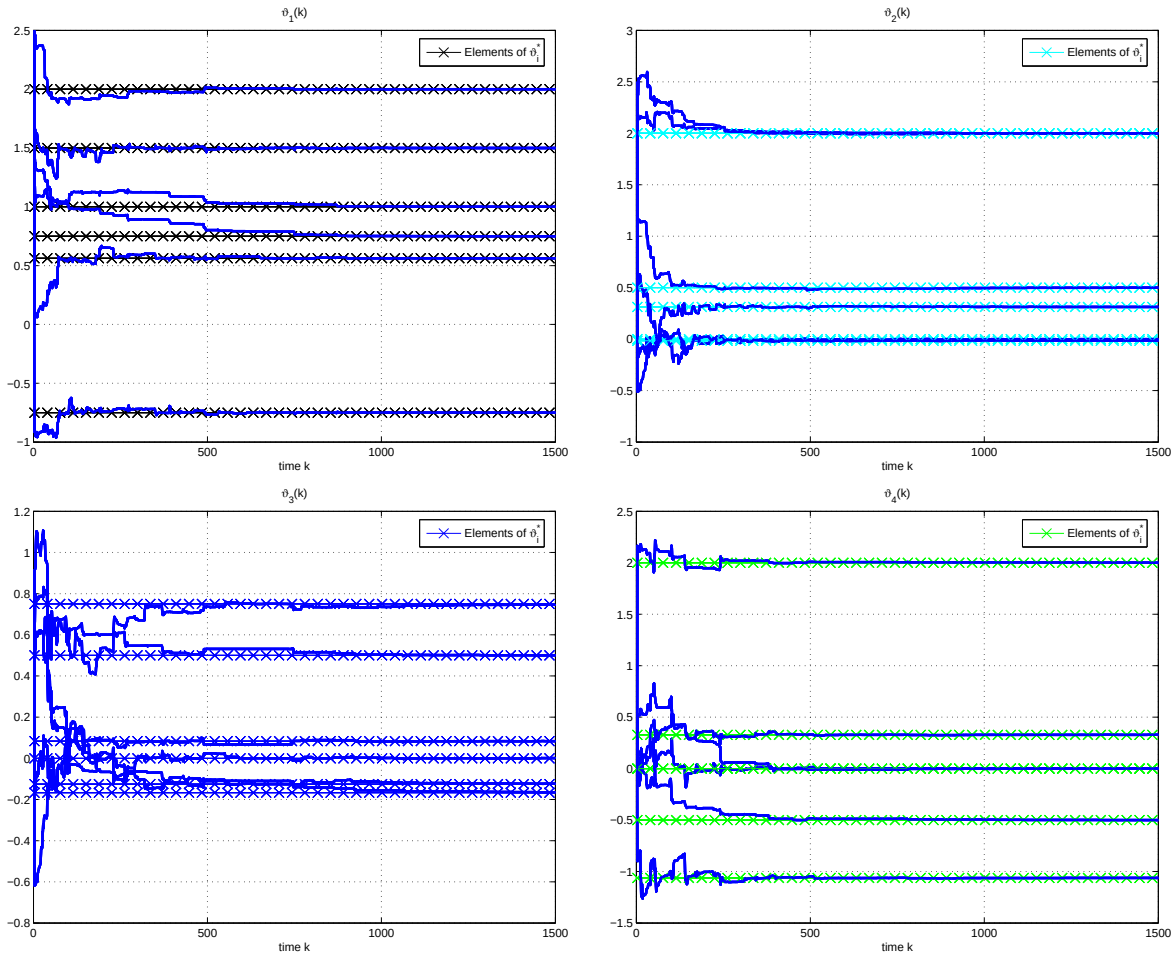

Figure 5.6: Convergence for the case $p = 6$.

model ◊ is very close to the same subsystem, the models +, ∗ and × did not move much yet.

- In Figure 5.10(c), □ has moved away from its subsystem a bit and now ◊ is closer. While × seems to have latched on to the subsystem in the left corner, as well as ∗ to the one in the center, + is stuck nowhere close to any subsystem.
- In Figure 5.10(d), □ is closest to the subsystem in the top right corner again, while ◊ moved away from it and + still is stuck in the middle.
- Finally, in Figure 5.10(e), a certain structure emerged and each model seems to have set out for an unique subsystem now.
- Figure 5.10(f) shows the convergence.

The main point here is that □ and ◊ initially compete for the subsystem in the top right corner, while + does not move a lot at all. It takes some time until □ "wins" this subsystem and ◊ moves to the one on the right. That is the system organizes itself until

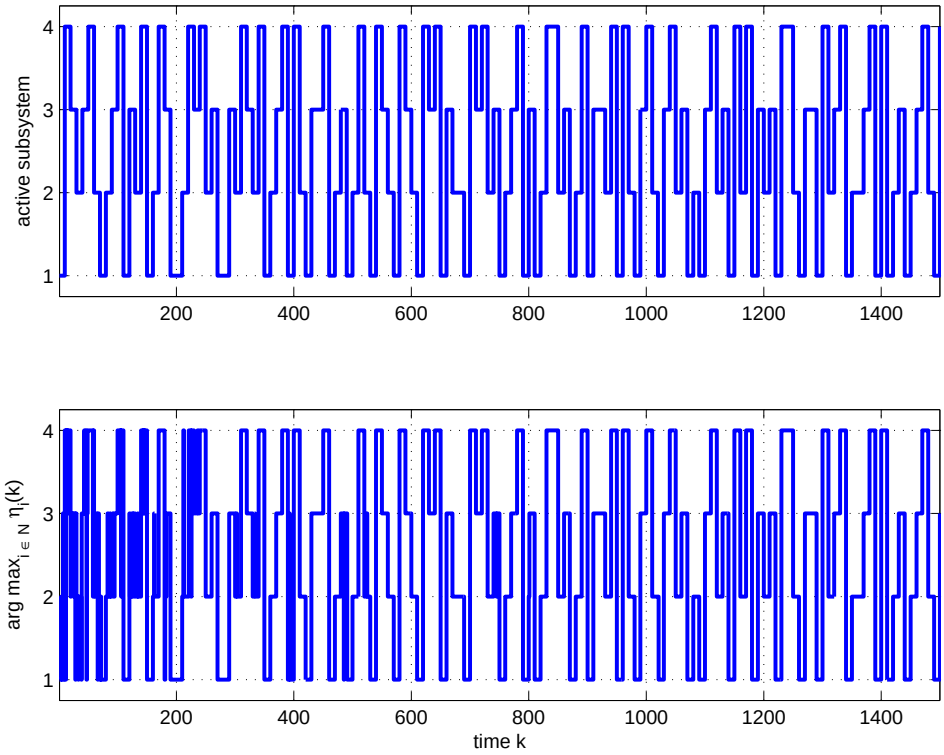

Figure 5.7: Switching of plant and models synchronizes for $p = 6$.

the final convergence can take place. Figure 5.11 consequently shows the phase portrait from time $k = 1450$ on. The models converge to their subsystems without getting close to each other anymore.

5.4.2 Difficulties in the convergence analysis

The simulation results of the preceding subsection were good evidence that in order to present a stability analysis of the presented algorithm, a lot of difficulties have to be overcome. We shall now quickly summarize some of those difficulties to conclude the treatment of the identification algorithm.

We are facing a nonlinear, time-variant $\nu(p+1)$-st order system (where p is the dimension of the parameter vector), constituted by Equations (5.1), (5.9) and (5.10) or in terms of the identification error e_i by

$$\begin{aligned} e_i(k) &= \varphi^T(k-d)[\vartheta_i(k-1) - \vartheta^*(k-d)] \\ \vartheta_i(k) &= \vartheta_i(k-1) - \eta_i(k)\gamma\varepsilon_i(k)\varphi(k-d) \\ \eta_i(k) &= \frac{e_i^{-2}(k)}{e_1^{-2}(k) + e_i^{-2}(k) + \cdots + e_\nu^{-2}(k)}. \end{aligned} \tag{5.11}$$

The main differences to the classical algorithms presented earlier in the thesis are on the

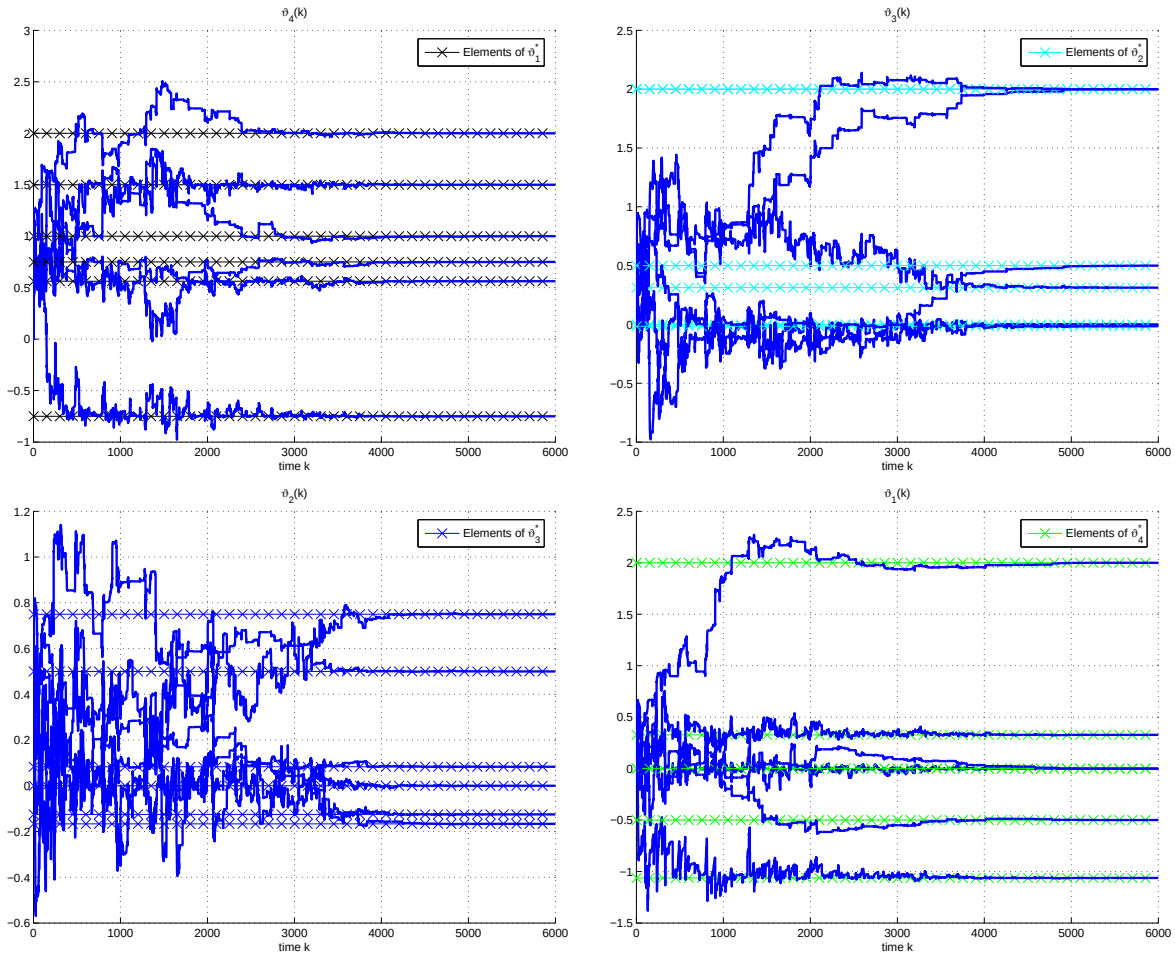

Figure 5.8: Convergence for the case $p = 6$ and models starting in a cluster.

one hand the explicit time-variance of ϑ^*, which was not considered in the theoretical analysis of the classical Multiple-Model schemes, and on the other hand that all equations are now *coupled* in an extremely nonlinear fashion by the factors η_i.

While the classical proofs of stability relied on the monotonic decrease of the Lyapunov function $\tilde{\vartheta}^T\tilde{\vartheta}$ (cf. the proof of Lemma 3.2, but also proofs for e.g. least-squares methods in [GS84, IF06]), application of such a function is prevented by the explicit time-variance of ϑ^*: It is not even possible to define the deviation variable $\tilde{\vartheta}_i$, since its definition changes with every switch in the parameter value. The examples have also shown that it is not determined solely by the initial conditions, which model will converge to each subsystem. But even if we somehow could decide beforehand that model i converges to subsystem j and defined $\tilde{\vartheta}_i = \vartheta_i - \vartheta_j^*$, the function $V(k) = \sum_{i\in N} \tilde{\vartheta}_i^T\tilde{\vartheta}_i$ will not decrease monotonically. As an example, the function V is plotted in Figure 5.12 for the examples from Figure 5.9. None of them decreases monotonically until the very end, when one could say that self-organization took place and convergence starts. While this of course does not negate the use of Lyapunov methods for the proof stability, it shows that

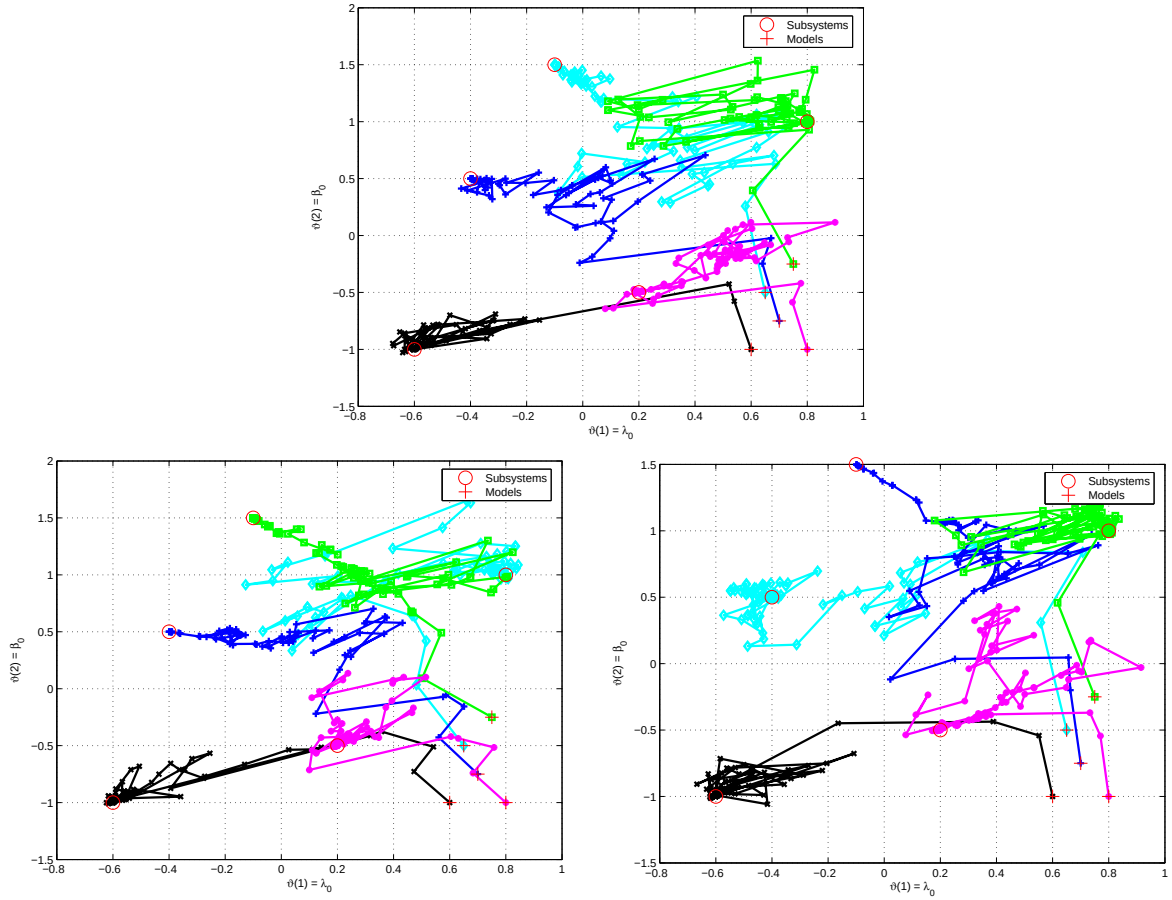

Figure 5.9: Models converge to different plants, even if the difference merely lies in the switching instances and sequence.

there lie enormous difficulties in trying to do so.

The analysis of individual models on the other side is prevented by the couplings. Although the dynamics of an individual model did not change much (note that from the point of view of a single model, η_i is merely a positive gain), the overall behaviour changed considerably, and this is due to the coupling. The analysis thus must consider the overall system, so probably little use can be made of earlier results on the individual models.

Concluding it can be stated that the stability of the overall system is extremely difficult and will probably not be possible in the classical framework of deviation variables and Lyapunov theory. In [FN06] it therefore is called for a entirely new mathematical framework for the analysis of the problem (or rather it was shown how the encountered problems could give rise to such a new framework). The idea is to find related dynamical systems with simpler dynamics such that the solution of the original system is enclosed by convex combinations of the vector fields of the simpler systems. For details see [FN06], but no further results have been reported on that issue yet.

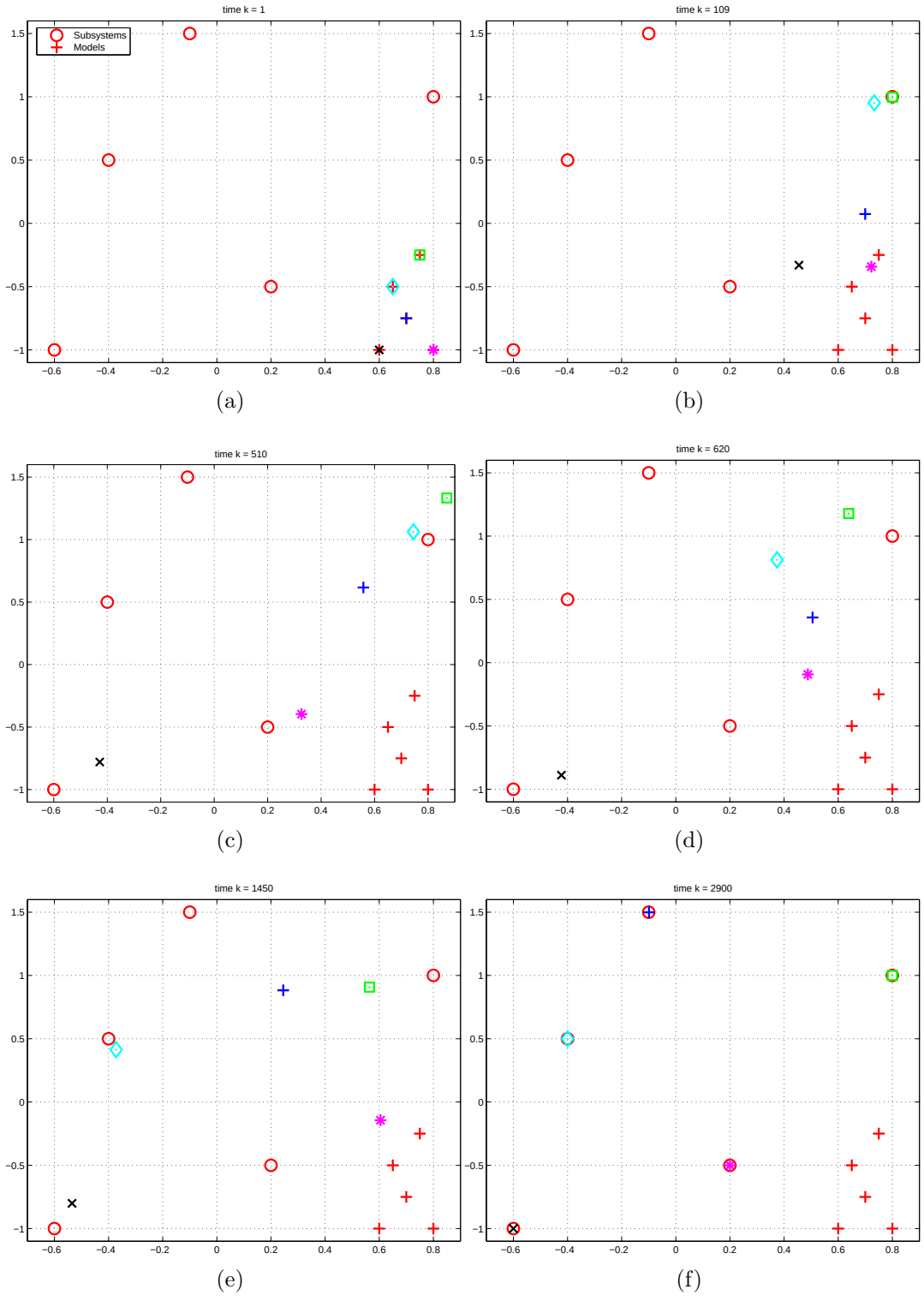

Figure 5.10: Snapshots of a phase portrait where the models started in a cluster.

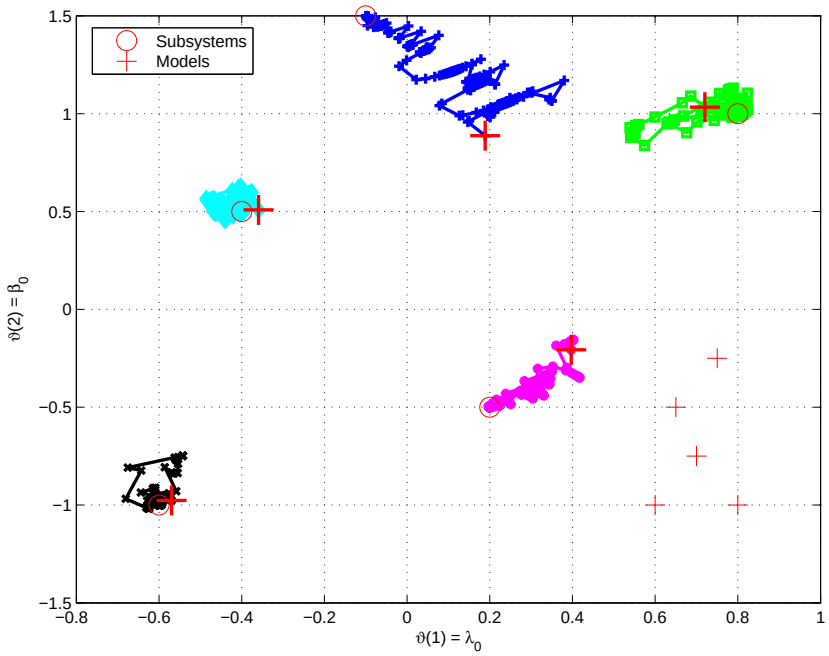

Figure 5.11: Phase portrait from $k = 1450$ on. The big + marks the values at $k = 1450$.

5.5 Simulations

Like the other chapters, we will conclude this chapter with simulation examples of the presented method. The first example will assume fast but periodic switching, in the second one the plant will switch arbitrarily between subsystems.

In both cases we have unstable subsystems with $d = 2$ and

$$\Theta^* = \left\{ \begin{bmatrix} -7/4 \\ -1/4 \\ 3/8 \\ 1 \\ 1 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 2 \\ -1/2 \\ -1/2 \\ 2 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 9/4 \\ -1 \\ -1 \\ 1/2 \\ 7/12 \\ 1/6 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ -1 \\ 2 \\ -3 \\ 1 \end{bmatrix} \right\} =: \{\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*\}. \tag{5.12}$$

The models start in the vicinity of the subsystems with $\|\vartheta_i(1) - \vartheta_i^*\| = 1$, the adaptive gain is $\gamma = 1$ and the reference signal is $y^*(k) = 10\sin(k/10)$. The only difference between the two experiments is the switching. In the first, the sequence of the subsystems is fixed and the switching periodic, in the second the sequence is random, as well as the switching.

Periodic switching in a fixed sequence. This represents the simplest case. The period is $k_p = 40$ and the sequence $1 \to 2 \to 3 \to 4 \to 1 \ldots$, i.e. $\theta^*(k) = \theta^*_{i^*(k)}$,

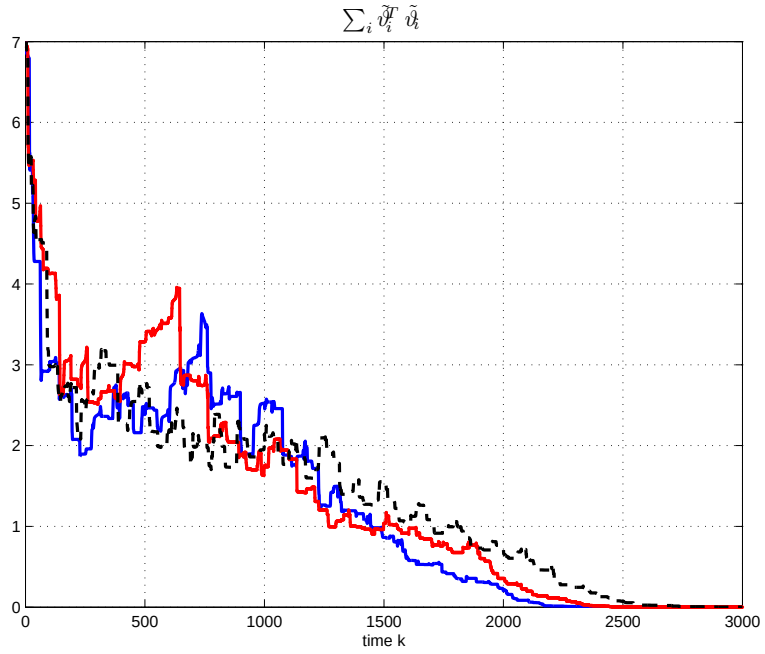

Figure 5.12: The Lyapunov function candidate $V = \sum_i \tilde{\vartheta}_i^T \tilde{\vartheta}_i$ does not decrease monotonically.

$i^*(k) = 1 + \lceil k/k_p \rceil \bmod 4$. Figure 5.13 shows the results. In the two lower plots the intervals I_c are shaded grey and it can be seen that the output does not follow the reference in exactly this intervals.

Random switching in a random sequence. Now the switching intervals are random between $d+1 = 3$ and 120 and the next subsystem is chosen randomly. The results are shown in Figure 5.14 and look qualitatively the same as the results in the periodic case.

In both cases, the models converge to the subsystems as well as the maximum value of η_i and the switching signal synchronize. Since the plots look qualitatively like Figures 5.6 and 5.7 they are omitted here.

Also in both cases, horrible transients appear in the initial phase, which disqualifies the use of the pure algorithm for direct practical application. It should be stressed though, that the transients are attenuated a lot if the models start closer to the subsystems and that the algorithm nevertheless yields a stable response and after the initial phase excellent performance, both of which could not have been achieved with any other available algorithm that does not make use of more initial information!

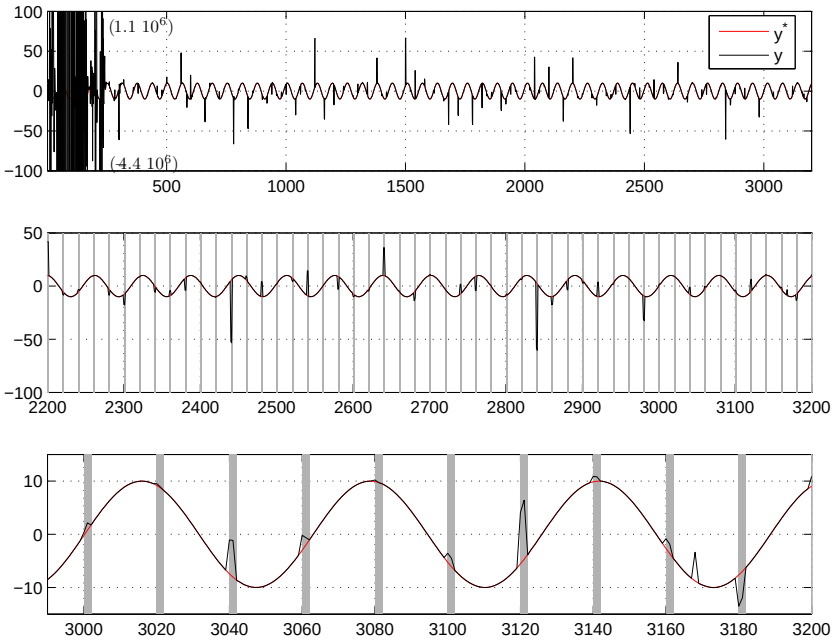

Figure 5.13: y^* and y in three different resolutions for the periodic example.

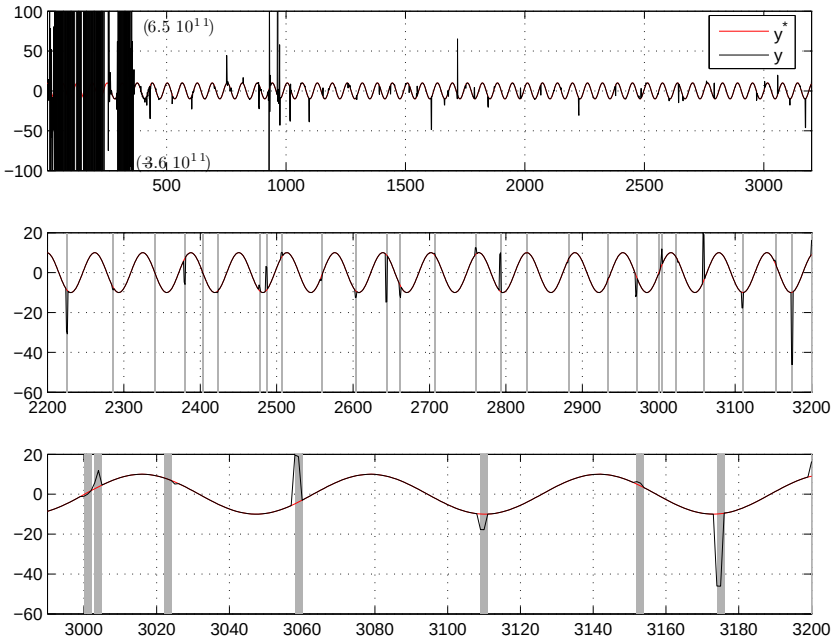

Figure 5.14: y^* and y in three different resolutions for the randomly switching example.

6 Concluding remarks

Recapitulation

The thesis provided a self-contained step-by-step treatment of the development that led from the classical single-model adaptive techniques to the novel algorithm first proposed in [FN06] and presented here in Chapter 5.

The main contribution of the thesis was to offer the first comprehensive description of the algorithm besides the conference paper [FN06], along with a discussion based on extensive simulation results. Additionally, a very broad background was presented by putting the algorithm in context with the already available adaptive algorithms. The motivation for this was to make the thesis a well-rounded piece of work that could be read and understood without resorting to additional literature and also to enable readers to see and judge for themselves, what the novelty and possible impact of the findings are.

The classical results of the first era of adaptive control, which made use of single models and did not take time-variance in account, were presented in Chapter 3. In particular, the projection algorithm was developed, including the proof of stability in Lemma 3.2, and subsequently teamed up with a simple control algorithm to yield a first Adaptive Control algorithm.

In Chapter 4 we have seen, where limitations of Adaptive Control lie and how considerable improvement could be achieved with the use of multiple models. The dominant approach of Multiple-Model Control with switching and tuning was detailed in Section 4.2.2. Remarkable thereby was that the proof of stability basically consisted of reducing the problem to the case of a single adaptive model.

Chapter 5 finally presented the "new way". It was pointed out that the novelty of the presented algorithm thereby lies in the fact that for the first time, adaptation with multiple models leads to simultaneous convergence to different points in parameter space and takes place independent of the length of the intervals of constant parameters. Unfortunately, stability of the nonlinear overall system, consisting of the new identification algorithm, a time-variant plant and the simple control algorithm of Section 3.2, could not be proven yet.

Outlook

Before a new theory can be confidently applied to practical problems, the stability properties have to be understood. Therefore it is natural that work on a proof of stability is ongoing. Parallel to that, the dual theory for continuous-time is developed. Actually, first simulations show that modification of the standard adaptive laws with the same factors $\eta_i(t) = \frac{e_i^{-2}(t)}{\sum e_j^{-2}(t)}$ leads to the same qualitative behaviour of the models converging to the subsystems, but work has only been rudimentary so far.

Results of this kind however meet a high demand. Especially for applications in aircraft control like treated as early as in [ACKP$^+$77] interest is still directed towards methods that can guarantee good performance in rapidly time-varying environments while relying on minimal information to save cost in e.g. sensors.

But besides usage for control and identification, the algorithm can be expected to have much broader applicability: Whenever multiple agents are required to perform self-organizing tasks with limited availability of external information such as distributed search or organization in swarms, the algorithm could turn out useful.

Appendix

The Appendix simply contains some assorted results and definitions which did not really fit anywhere in the prequel but were felt to should be accessible to the reader without referring to secondary literature, so it contains not even a rudimentary treatment of any concepts or theories.

Stability, Observability

Definition A.1. A system is called *observable*, if the knowledge of the inputs and outputs over some finite interval is sufficient to reconstruct the initial state.

Theorem A.2 (Observable pair, observability matrix)**.** *A LTI system in state-space form as in* (2.6) *is observable if and only if* (A, c^T) *is an* observable pair*, which is the case if and only if the* observability matrix

$$\mathcal{O} = \begin{bmatrix} c^T \\ c^T A \\ \vdots \\ c^T A^{n-1} \end{bmatrix}$$

has rank n*.*

It is worth mentioning that the proof is more or less given in the derivation of Equation (2.9).

Throughout the thesis we preferred to use the backward shift operator q because it is more convenient in terms of implementation. For stability considerations however, it turns out to be preferable to use the forward shift operator z, such that $zy(k) = y(k+1)$. The following lemma thus makes use of z instead of q, it is easy to translate both forms into each other by simply multiplying both sides with the appropriate negative power of z (q) and substitute z^{-1} by q and vice versa.

Lemma A.3 (BIBO-Stability)**.** *If we have an LTI-system*

$$A(z)y(k) = B(z)u(k), \tag{A-1}$$

where $A(z)$ *and* $B(z)$ *are polynomials in* z *it is stable if*

1. *All poles, that is all zeros of $A(z)$ lie inside or on the closed unit disc* [i] *($|z| \leq 1$ $\forall z \in \mathcal{C} : A(z) = 0$),*
2. *All poles with $|z| = 1$ are also zeros of $B(z)$, i.e. they are uncontrollable,*
3. *All poles with $|z| = 1$ correspond to* Jordan *blocks of dimension 1 in a state-space representation.*

Sketch of a proof. That Item 1 with a strict inequality (i.e. all roots lie inside the unit circle) is a sufficient condition is common knowledge and can be found in any book on discrete-time systems, e.g. [ÅW97]. If the poles are not controllable, these modes are not excited by the input but only by their initial conditions. As long as the corresponding Jordan block has dimension 1, this can only cause an undamped oscillation or a constant added to the output of the controllable part. □

Bezout identity

The following result has a more general counterpart in terms of Euclidean rings but the common formulation in terms of polynomials is sufficient to derive the d-step ahead predictor form in Section 2.2.

Theorem A.4 (Bezout Identity)**.** *Let $M(s)$, $N(s)$ be two polynomials of order m and n, respectively. Then, there exist polynomials $Q(s)$ with $\deg Q \leq n - 1$ and $R(s)$ with $\deg R \leq m - 1$ such that*

$$M(s)Q(s) + N(s)R(s) = 1 \tag{A-2}$$

if and only if $M(s)$ and $N(s)$ are coprime.

[i] Strictly speaking, since z was introduced as an operator, this is formally incorrect and we should have introduced it as the z-Transform variable. However, it makes no difference here and we accept this slight inconsistency.

Bibliography

[ACKP+77] M. Athans, D. Castañon, D. Keh-Ping, C.S. Greene, H.L. Wing, N.R. Sandell, and A.S. Willsky, *The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method – Part I: Equilibrium flight*, IEEE Transactions on Automatic Control **22** (1977), no. 5, 768–780.

[ÅW97] Karl Åström and Björn Wittenmark, *Computer controlled systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1997.

[FB86] M. Fu and B.R. Barmish, *Adaptive stabilization of linear systems via switching control*, IEEE Transactions on Automatic Control **31** (1986), no. 12, 1097–1103.

[FN06] M.J. Feiler and K.S. Narendra, *Simultaneous identification and control of time-varying systems*, IEEE Conference on Decision and Control (San Diego), 2006, pp. 1093–1098.

[GS84] Graham C. Goodwin and Kwai Sang Sin, *Adaptive filtering, prediction and control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1984.

[IF06] Petros Ioannou and Barış Fidan, *Adaptive control tutorial*, SIAM, Philadelphia, PA, USA, 2006.

[Lai76] D.G. Lainiotis, *Partitioning: A unifying framework for adaptive systems, I: Identification and II: Control*, Proceedings of the IEEE **64** (1976), no. 8, 1126–1143 and 1182–1197.

[Mag65] D.T. Magill, *Optimal adaptive estimation of sampled stochastic processes*, IEEE Transactions on Automatic Control **10** (1965), no. 4, 434–439.

[Mar86] Bengt Martensson, *Adaptive Stabilization*, PhD dissertation, Lund Institute of Technology, Department of Automatic Control, 1986.

[MGHM92] R.H. Middleton, G.C. Goodwin, D.J. Hill, and D.Q. Mayne, *Design issues in adaptive control*, IEEE Transactions on Automatic Control **33** (1992), no. 1, 1343–1354.

[NA89] Kumpati S. Narendra and Anuradha M. Annaswamy, *Stable adaptive systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[NB97] K.S. Narendra and J. Balakrishnan, *Adaptive control using multiple models*, IEEE Transactions on Automatic Control **3** (1997), no. 42, 171–187.

[NBC95] K.S. Narendra, J. Balakrishnan, and M.K. Ciliz, *Adaptation and learning using multiple models, switching, and tuning*, IEEE Control Systems Magazine **15** (1995), no. 42, 37–51.

[NX00] K.S. Narendra and Ch. Xiang, *Adaptive control of discrete-time systems using multiple models*, IEEE Transactions on Automatic Control **45** (2000), no. 9, 1669–1685.

[SD72] G.N. Saridis and T.K. Dao, *A learning approach to the parameter self organizing control problem*, Automatica **8** (1972), 589–597.

[Xia00] Cheng Xiang, *Adaptive control of discrete-time systems using multiple models*, PhD dissertation, Yale University, Center for Systems Science, 2000.