Akademie věd České republiky
Ústav teorie informace a automatizace

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Hong Gao, Miroslav Kárný

Bayesian Approximate On-line Identification and Adaptive Control of Large-state-space and High-order Markov Chains

No. 1957 August 1999

ÚTIA AV ČR, P. O. Box 18, 182 08 Prague, Czech Republic
Fax: (+42)(2)688 4903
E-mail: utia@utia.cas.cz

Acknowledgements

This work is financially supported by GAAV ČR, Grant No. A2075603, A207606, GA ČR 102/98/P059, 102/97/0118, which are gratefully acknowledged.

Contents

Symbols and Notations

Mathematical symbols

$p(\cdot)$	probability
$p(\cdot\vert\cdot)$	conditional probability
Y_t	continuous valued output at discrete time t
U_t	continuous valued input at discrete time t
y_t	discrete valued output at discrete time t
u_t	discrete valued input at discrete time t
$d_t = (y_t, u_t)$	data at discrete time t
$D_t = (d_1, d_2, \cdots, d_t)$	data sequence from time 1 to t
d_0	prior information about system input and output
$\phi_{t-1} = (d_{t-1}, \cdots, d_{t-m})$	data sequence from time $t-m$ to $t-1$
$\varphi_t = (u_t, \phi_{t-1})$	regressor
x^*	state space of stochastic sequence x_t
m	memory length or model order
m_{x^*}	dimension of the state space x^*
Θ	unknown transition probability matrix with entry $\theta_{i\vert j}$
$\propto$	proportionality
$n(t)$	sufficient statistic matrix with entry $n_{i\vert j}(t)$
$B(\cdot)$	Beta function
$\Gamma(\cdot)$	Gamma function
$\mathcal{D}(\cdot \parallel \cdot)$	Kullback-Leibler distance
$y_t^{[q]}$	discrete valued output corresponding to the qth quamtizer
$u_t^{[q]}$	discrete valued input corresponding to the qth quantizer
$y_t^{[l]}$	discrete valued output corresponding to the lth subsystem
$u_t^{[l]}$	discrete valued intput corresponding to the lth subsystem
H	control horizon
L	loss function
$p(\Theta)$	prior probability density function of the unknown parameter Θ
$E[\cdot]$	expectation

Abbreviations

MC	Markov chain
CMC	controlled Markov chain
SISO	single input and single output
ARX	Auto-Regression models with eXternal inputs
AD	aggregation-disaggregation
tpm	transition probability matrix
pdf	probability density function
MAP	maximum a posterior probability
NN	neural network

Chapter 1

Introduction

This thesis concerns the questions of approximate identification and adaptive control of stochastic systems modelled by finite controlled Markov chains (CMC). The treatment of these questions is unified by adopting the viewpoint of one who must make decision under uncertainty. The area of making decision under uncertainty is vast. This chapter gives only a brief survey of this field. The stress is put on Markov decision process.

The outline of this chapter is as follows. Section 1.1 gives a definitive description of the field of making decision under uncertainty. Section 1.2 briefly summarizes the development of this field. The motivation of the thesis is mentioned in Section 1.3. Section 1.4 presents the significance of the thesis. Section 1.5 explains why Bayesian approach is adopted throughout the thesis. The last section serves as a thesis layout.

1.1 Making Decision under Uncertainty

We are concerned with controlling a dynamic system whose behavior is not completely foreseeable. That is, we deal with a system whose future evolution is not completely determined by its present state and future control actions as is the case with deterministic systems. The uncertainty associated with the future behavior can be quantified in several ways. The adopted approach assumes that this quantification is probabilistic one. We say that we are dealing with stochastic dynamic systems. The dynamics of such systems relating quantities with a finite amount of values can be described by controlled Markov chains. This thesis is dedicated to decision making using *controlled Markov chains.*

In practice the analyst faces two different issues. The first issue concerns the use of the information obtained from the observed outputs to reduce the uncertainty about the system behavior. Second, the analyst has to assess the effect on the system behavior of alternative control policies and to select the most desirable policy. The first concern leads to the problem of *state estimation* and to the problem of *system identification*. The second is formulated as a problem of *stochastic control*. This thesis provides an account of two aspects of decision-making: identification and control, especially for the single input and single output (SISO) systems that can be modelled by controlled Markov chains.

1.2 State of the Art

Since its origins in the 1940s [1, 2, 3, 4], the subject of decision making under uncertainty has grown into a ramified area with applications in several branches of engineering and in those areas of the social sciences concerned with policy analysis and prescription [5]. The theoretical approaches to the subject derive from the invention of dynamic programming in the 1950s and rich body of research in statistical time series analysis [6, 7].

Although theoretically appealing, these approaches found little practical use since they demanded a computing capacity that was too expensive at that time. The tremendous advances in information technology provided the motivation for subsequent impulses that advanced the subject. Recent years have witnessed an explosion of work aimed at applications in signal processing, control, operations research and economics.

Markov decision process is a very popular and important decision making method not only because very many systems of interest can be modelled by Markov models, but also because the mathematical theory and computational techniques are in a much better shape than for non-Markov models. It has been applied in many fields that have adopted Markov chains as mathematical models.

1.3 Motivation

Adaptive predictors and controllers in signal processing and control are mostly based on autoregressive models with external inputs (ARX). A lot of energy has been spent in this direction and a lot has been gained, e.g. [8, 9, 10, 11, 12]. Now ARX models are still widely used in various systems, e.g. [13, 14, 15]. There are, however, applications for which the limitations of ARX models might be significant. Specifically, ARX models are linear and as such do not respect

- process nonlinearities;
- limited range of treated signals;
- digitized (discrete) data.

When looking for an alternative universal class of models, finite order Markov chains seem to be promising alternative.

Markov chain models describe systems by transition probabilities which become finite-dimensional tables since the number of argument values is finite. Considering the transition table as a set of unknown parameters, it is easy to find its Bayesian predictor [16] and controller [17]. This property together with universality of the model – it covers all time invariant digitized discrete time systems with finite memory – make the Markov chain model an ideal tool for adaptive prediction and control.

However, exponential burst of the transition-table dimensionality with increasing dimension of the state space and memory length restricts applicability of this model. This problem motivates the search for a feasible approximation.

The main theme of this thesis is to fight with the curse of dimensionality inherent to Markov chains. A practical approximation methodology is proposed both for identification and adaptive control of large Markov chains. It is tested on several simulated and real systems. The present experiments confirm feasibility of this methodology.

1.4 Approximation

Many methodologies for aggregation-disaggregation (AD) in large Markov chains have been proposed in recent years [18]. Some methodologies have been used for economical or social systems. However, the referred AD algorithms are not suitable for on-line adaptive prediction and control. A novel practical AD method is needed in order to apply MC models to wider fields, especially adaptive systems working permanently in real time. Almost no result can be found in this respect in the literature.

A promising approximation methodology is proposed and studied in this thesis. Because cardinalities of state space and memory length are the two fundamental reasons to cause the large dimensionality, the basic idea of this methodology is to downsize these two dimensions respectively. The detailed procedure is as follows.

We split the original model into a set of simple models with small state space and short memory length first; then these simple models are estimated and corresponding control design made; lastly, the inputs proposed by particular controllers are mixed into the final action applied.

1.5 Bayesian Paradigm

Throughout this thesis, we use Bayesian approach for identifying and controlling the real systems modelled by Markov chains.

The Bayesian approach gives a related but conceptually different treatment of the parameter estimation problem compared with traditional Least Squares or Maximum Likehood point estimation methods [19, 20].

It describes the random behavior of data through their probability distribution and even the uncertainty of assigning such a distribution in terms of probability. Probability can thus appear in double roles. On one hand, as a relative frequency of possible outcomes of a certain experiment in a long run. On the other hand, as a measure of uncertainty of unknown quantities. Bayesian identification of Markov chains is of the most simple form when taking transition probability matrix as unknown parameters as only simple counting of frequency that all possible situations occur is needed.

The pros and cons of the Bayesian paradigm can be well illustrated on the case of stochastic adaptive control. Suppose we are to control a dynamic system that depends on some unknown parameter Θ. We have two options then. Either we rely on a point estimate of Θ, or we take the uncertainty of Θ into account. In the latter case, the expectation in a cost function applies to both the stochastic behavior of the system and the uncertainty of the parameter Θ. This converts the original problem into a hyperproblem that has no more unknowns. Its information state is formed by the probability density function of the original state and parameter Θ conditional on the observed data.

The beauty of the result is that it looks as if all uncertainty vanished. As soon as the prior density is set, the state evolves in a definite way, governed by the laws of probability theory.

1.6 Thesis Layout

This thesis is divided into 10 chapters.

The first chapter is introductory. It gives a brief description of the field we are moving and the thesis outline.

The second chapter summarizes the mathematical background required in this thesis so that readers can achieve a better understanding of the rest part of the thesis. It presents some basic definitions and notations of Markov chains. Moreover, the notion of a system model is revised from Bayesian viewpoint.

Chapter 3 to Chapter 5 deal with the problems of Bayesian identification and prediction of Markov chains. Chapter 3 lays a theoretical foundation for the following two chapters by presenting the theory of Bayesian identification and prediction of Markov chains. This appealing theory fails in application because of dimension drawback. Therefore, Chapter 4 proposes an approximate prediction methodology in order to apply MCs to on-line mode. In Chapter 5, the performance of the proposed approximate prediction methodology is tested in both simulated and real systems.

Chapter 6 to Chapter 9 deal with the problems of Bayesian adaptive control of Markov chains. Chapter 6 theoretically formulates the Bayesian adaptive control problem. Furthermore, one simple example (a controlled tossing of a coin) is presented to illustrate complexity and basic features of Bayesian adaptive control problem. In order to combat complexity of Bayesian adaptive control problem, Chapter 7 presents an alternative design formulation, namely fully probabilistic control design. Based on this control design, the above approximate prediction methodology is extended to the adaptive control case in Chapter 8. This methodology is tested on both simulated and real systems in Chapter 9.

In the last chapter, the thesis is concluded by presenting a summary of the results obtained and suggestions for future research.

Chapter 2

Markov Chain and Uncertainty

This chapter summarizes the mathematical background required in this thesis. The purpose is to remind readers of basic notions and results about Markov chains and uncertainties so that this thesis is largely seft-contained. Moreover, a stochastic model of a continuous state system is revised from Bayesian viewpoint.

The outline of this chapter is as follows. Section 2.1 presents some basic definitions and notations of Markov chains. The basic relations on uncertainty are summed up in Section 2.2. Section 2.3 revises the notion of a system model from Bayesian viewpoint. Lastly, Section 2.4 summarizes the main results of this chapter.

2.1 Markov Chain

The concept of Markov chain was introduced by Russian mathematician Andrei Andreievich Markov in the early 20th century. An introduction to controlled Markov chain models can be found in [21]. Finite Markov chains are thoroughly treated in [22, 23]. In this section, we present various notations and definitions of Markov chains used in this thesis.

• **Markov chain**

Let $\{x_t, t = 0, 1, 2, ...\}$ be a sequence of random variables which assume values in a discrete (finite or countable) state space x^*. We say that $\{x_t, t = 0, 1, 2, \cdots,\}$ is a *Markov chain* if

$$
\begin{aligned}
& p_t(x_t = j_t | x_{t-1} = j_{t-1}, x_{t-2} = j_{t-2}, \cdots, x_0 = j_0) \\
& = p_t(x_t = j_t | x_{t-1} = j_{t-1})
\end{aligned}
$$

for $t \geq 1$ and $j_t \in x^*$ where $p(\cdot)$ is probability.

• **mth order Markov chain**

If there holds that

$$p_t(x_t = j_t | x_{t-1} = j_{t-1}, x_{t-2} = j_{t-2}, \cdots, x_0 = j_0)$$

$$= p_t(x_t = j_t|x_{t-1} = j_{t-1}, x_{t-2} = j_{t-2}, \cdots, x_{t-m} = j_{t-m}),$$

we say that $\{x_t, t = 0, 1, 2, \cdots,\}$ is a *mth order Markov chain.*

• **stationary Markov chain**
If $p_t(x_t = i|x_{t-1} = j), i, j \in x^*$ does not depend on t, i.e.,

$$p_t(x_t = i|x_{t-1} = j) = p(x_t = i|x_{t-1} = j),$$

we call this Markov chain *stationary.*

• **transition probability matrix**
For a stationary Markov chain, we write $p_{i|j} = p(x_t = i|x_{t-1} = j)$ and refer to it as a transition probability. We call the matrix P with (i, j)th component $p_{i|j}$ the *transition probability matrix* (tpm).

The matrix P has the property that every element is non-negative, and the sum of the elements in every column is 1.

2.2 Basic Relations on Uncertainty

In this section, we sum up some basic relations on uncertainty whose application leads to solutions to all identification and control problems we deal with.

Given three uncertain quantities a, b and c, hold the following three fundamental relations.
• **chain rule**

$$p(a, b|c) = p(a|b, c)p(b|c)$$

where $p(\cdot|\cdot)$ represents conditional probability.
• **marginalization**

$$p(b|c) = \int p(a, b|c)da$$

• **Bayes formula**

$$p(a|b, c) = \frac{p(b|a, c)p(a|c)}{\int p(b|a, c)p(a|c)da} \propto p(b|a, c)p(a|c)$$

where the symbol $\propto$ stands for proportionality, i.e., equality up to a normalizing factor.

The integral taken over a^* is converted into a sum whenever a is a discrete-valued quantity.

Bayes formula is the heart of Bayesian statistics. It can be derived from the above two relations.

2.3 Markov Chain Model

In this section we reexamine a system model from Bayesian viewpoint.

Consider a stochastic system S working at discrete time $t = 1, 2, \cdots$ with single continuous valued output Y_t and single continuous valued input U_t. The corresponding discrete valued output $y_t \in y^* \equiv \{i\}_{i=1}^{m_{y^*}}$, $m_{y^*} < \infty$ and input $u_t \in u^* \equiv \{j\}_{j=1}^{m_{u^*}}$, $m_{u^*} < \infty$ sequences can be obtained via analog-digital and digital-analog converters.

Denote the discrete data pair as $d_t = (u_t, y_t)$. The sequence of data observed up to time t is denoted as

$$D_t = (d_t, d_{t-1}, \ldots, d_1).$$

Assume that we have a prior information of the system d_0 and the present data only depend on m step previous data. m is called *memory length* or *model order* that is assumed finite. Therefore, in the considered time interval T the discretized stochastic system S can be fully described by a conditional probability function $p(D_T|d_0)$. Applying the chain rule to $p(D_T|d_0)$ gives

$$\begin{aligned}
p(D_T|d_0) &= p(d_T, d_{T-1}, \cdots, d_1|d_0) \\
&= \prod_{t=1}^{T} p_t(d_t|D_{t-1}) = \prod_{t=1}^{T} p_t(y_t, u_t|D_{t-1}) \\
&= \prod_{t=1}^{T} p_t(y_t|u_t, D_{t-1}) p_t(u_t|D_{t-1}) \\
&= \prod_{t=1}^{T} p_t(y_t|u_t, d_{t-1}, \cdots, d_{t-m}) p_t(u_t|d_{t-1}, \cdots, d_{t-m}) \qquad (2.1)
\end{aligned}$$

The factors in (2.1) have the following interpretation. The conditional probability distribution

$$p_t(u_t|d_{t-1}, \cdots, d_{t-m}) \qquad (2.2)$$

describes the transformation, in general stochastic, by which the input u_t is determined on the basis of the known past history of the system. The set of functions (2.2) is actually the control strategy. The remaining factor in (2.1), i.e.,

$$p_t(y_t|u_t, d_{t-1}, \cdots, d_{t-m}) \qquad (2.3)$$

describes, for each t, the dependence of the output y_t on the known past history including the last input. It is the most general description of the system. We call it **system model**. It is used to design a control strategy or to forecast the output for a given control strategy.

The family of possible Markov processes, each member of which corresponds to a fixed sequence of controls or decision rules, is called a **controlled Markov chain**

(CMC). Therefore, the discretized stochastic system S is modelled by a mth order controlled Markov chain.

If the system S is time-invariant, it can be modeled by a stationary MC, the system model (2.3) becomes

$$\begin{aligned} p_t(y_t|u_t, d_{t-1}, \cdots, d_{t-m}) &= p(y_t|u_t, d_{t-1}, \cdots, d_{t-m}) \\ &= p(y_t|\phi_{t-1}, u_t) = p(y_t|\varphi_t) \end{aligned}$$

where $\phi_{t-1} = (d_{t-1}, d_{t-2}, \cdots, d_{t-m}),\ \varphi_t = (u_t, \phi_{t-1}),\ m < \infty$. φ_t corresponds to the regressor in an ARX model.

2.4 Summary

In this chapter we sum up the main definitions and results in Markov chains and uncertainties which lead to the results in the following chapters. The main contribution of this chapter is that the discretized stochastic systems are modelled by the controlled Markov chains via Bayesian viewpoint.

Chapter 3

Bayesian Identification and Prediction

The purpose of system identification, considered throughout this chapter, is its potential use for prediction and control of an uncertain system. Prediction is the most important prerequisite of a rational control. This chapter presents the theoretical solution to identification and prediction of Markov chains via Bayesian approach.

The outline of this chapter is as follows. Section 3.1 deals with parameter estimation of MCs via Bayesian statistics. Section 3.2 presents Bayesian prediction based on MCs. The problem of "slowly varying" parameters is addressed by using exponential forgetting technique in Section 3.3. Section 3.4 explains how to select the best model order via Bayesian approach. Lastly, Section 3.5 summarizes the main results of this chapter.

3.1 Bayesian Estimation

Consider a stochastic system modelled by a mth order stationary Markov chain $p(y_t|\varphi_{t-1})$. Very often this system model is unkown or partially unknown to us. It means there are some unknown parameters in the transition probability matrix. In this case, this system is described by a parameterized system model $p(y_t = i|\varphi_t = j, \Theta) = \theta_{i|j}$ where $\Theta \in \Theta^*$ is the unknown transition probability matrix with entry $\theta_{i|j}$, $\Theta^* \equiv \{\theta_{i|j} \geq 0, \sum_{i \in y^*} \theta_{i|j} = 1, \forall j \in \varphi^*\}$, j is a scalar integar coded from $(y_{t-1} = j_1, \cdots, y_{t-m} = j_m, u_t = k_0, u_{t-1} = k_1, \cdots, u_{t-m} = k_m)$.

We adopt Bayesian approach to estimate this transition probability matrix Θ. Bayesian estimation is a probability-based estimation [16], which interprets the unknown parameter Θ as a random variable. Therefore, it is natural to describe its uncertainty through its posterior probability density function (pdf) conditional on the observed data D_t.

Suppose we have a certain prior information about the unknown parameter Θ,

$p(\Theta)$. The posterior pdf $p(\Theta|D_t)$ can be found by applying the Bayes formula.

$$p(\Theta|D_t) = \frac{p(y_t|u_t, D_{t-1}, \Theta)p(\Theta|u_t, D_{t-1})}{\int p(y_t|u_t, D_{t-1}, \Theta)p(\Theta|u_t, D_{t-1})d\Theta} \tag{3.1}$$

under adopted natural conditions of control [16],

$$p(\Theta|u_t, D_{t-1}) = p(\Theta|D_{t-1}). \tag{3.2}$$

It means the sole decision u_t can not bring any additional information about Θ, i.e., Θ is independent of u_t when conditioned on D_{t-1}. Therefore,

$$\begin{aligned} p(\Theta|D_t) &= \frac{p(y_t|u_t, D_{t-1}, \Theta)p(\Theta|D_{t-1})}{\int p(y_t|u_t, D_{t-1}, \Theta)p(\Theta|D_{t-1})d\Theta} \\ &\propto p(y_t|u_t, D_{t-1}, \Theta)p(\Theta|D_{t-1}) \end{aligned} \tag{3.3}$$

This is Bayesian recursive estimation of an unknown parameter Θ. It bases on the parametrized system model $p(y_t|u_t, D_{t-1}, \Theta)$ evaluated at measured data. The recursion starts at time $t = 0$ when $p(\Theta|D_0) = p(\Theta)$ expresses the prior information about the unknown parameter Θ.

Applying (3.3) recursively gives

$$\begin{aligned} p(\Theta|D_t) &= \frac{\prod_{\tau=1}^{t} p(y_\tau|u_\tau, D_{\tau-1}, \Theta)p(\Theta)}{\int \prod_{\tau=1}^{t} p(y_\tau|u_\tau, D_{\tau-1}, \Theta)p(\Theta)d\Theta} \\ &= \frac{\prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, \Theta)p(\Theta)}{\int \prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, \Theta)p(\Theta)d\Theta} \qquad (3.4) \\ &\propto L_\Theta(t)p(\Theta) \qquad (3.5) \end{aligned}$$

This is Bayesian one shot estimation of an unknown parameter Θ.
$L_\Theta(t) = \prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, \Theta)$ is the likelihood function, i.e., product of the parametrized models at measured data.

Now let's apply Bayesian estimation to Markov chains.

$$\begin{aligned} p(\Theta|D_t) &= \frac{\prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, \Theta)p(\Theta)}{\int \prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, \Theta)p(\Theta)d\Theta} \\ &= \frac{\prod_{\tau=1}^{t} \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\delta_{y_\tau,i}\delta_{\varphi_\tau,j}} p(\Theta)}{\int \prod_{\tau=1}^{t} \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\delta_{y_\tau,i}\delta_{\varphi_\tau,j}} p(\Theta)d\Theta} \\ &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\sum_{\tau=1}^{t} \delta_{y_\tau,i}\delta_{\varphi_\tau,j}} p(\Theta)}{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\sum_{\tau=1}^{t} \delta_{y_\tau,i}\delta_{\varphi_\tau,j}} p(\Theta)d\Theta} \end{aligned}$$

$$= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)} p(\Theta)}{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)} p(\Theta) d\Theta} \tag{3.6}$$

where $y_\tau \in y^* \equiv \{i\}_{i=1}^{m_{y^*}}$, $\varphi_\tau \in \varphi^* \equiv \{j\}_{j=1}^{m_{\varphi^*}}$, $\delta_{x,y} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$ and $n_{i|j}(t) = \sum_{\tau=1}^{t} \delta_{y_\tau,i}\delta_{\varphi_\tau,j}$ which denotes the number of the observed situations

$$y_\tau = i, \ \varphi_\tau = j, \ \tau = 1, \ldots, t. \tag{3.7}$$

If we select the prior information $p(\Theta)$ in a conjugate form

$$p(\Theta) \propto \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(0)-1} \tag{3.8}$$

with some $n_{i|j}(0) > 0$, the posterior density function $p(\Theta|D_t)$ of Θ is a function of the sufficient statistic matrix $n(t)$ with entry $n_{i|j}(t)$.

$$p(\Theta|D_t) = \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)+n_{i|j}(0)-1}}{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)+n_{i|j}(0)-1} d\Theta} \tag{3.9}$$

In order to simplify the notation, we include $n_{i|j}(0)$ into $n_{i|j}$. Therefore, we can rewrite (3.9) as

$$\begin{aligned} p(\Theta|D_t) &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1}}{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1} d\Theta} \\ &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1}}{B(n(t))} \qquad (3.10) \\ &\propto \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1} \qquad (3.11) \end{aligned}$$

where $B(\cdot)$ is a multivariate Beta function [24].

Notice that the posterior density funtion $p(\Theta|D_t)$ in (3.11) is determined only by the sufficient statistic $n(t)$. Therefore, Bayesian estimation of Markov chains is nothing more than a simple counting of $n(t)$ according to (3.7). The recursive version is

$$n_{i|j}(t) = n_{i|j}(t-1) + \delta_{y_t,i}\delta_{\varphi_t,j}. \tag{3.12}$$

The recursion starts at the chosen $n(0)$. The sufficient statistic $n(t)$ evolves by counting the occurrence number of the situation (y_τ, φ_τ) for $\tau < t$.

3.2 Bayesian Prediction

Bayesian prediction $p(y_{t+1}|u_{t+1}, D_t)$ can be easily obtained based on Bayesian estimation. According to marginalization rule, Bayesian formula and natural conditions of control,

$$\begin{aligned} p(y_{t+1}|u_{t+1}, D_t) &= \int p(y_{t+1}, \Theta|u_{t+1}, D_t) d\Theta \\ &= \int p(y_{t+1}|u_{t+1}, D_t, \Theta) p(\Theta|u_{t+1}, D_t) d\Theta \\ &= \int p(y_{t+1}|\varphi_{t+1}, \Theta) p(\Theta|D_t) d\Theta. \end{aligned} \tag{3.13}$$

It means Bayesian prediction can be obtained from the parametrized system model and parameter estimate.

Let us compute it for MC case. Applying (3.10) to (3.13) gives

$$\begin{aligned} p(y_{t+1}|u_{t+1}, D_t) &= \frac{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\delta_{y_{t+1},i}\delta_{\varphi_{t+1},j}} \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1} d\Theta}{B(n(t))} \\ &= \frac{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t+1)-1} d\Theta}{B(n(t))} \\ &= \frac{B(n(t+1))}{B(n(t))} \end{aligned} \tag{3.14}$$

The definition of Beta function, recursion (3.12) and the known property of Γ function $\Gamma(x+1) = x\Gamma(x)$ imply

$$\begin{aligned} B(n(t+1)) &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t+1)]}{\Gamma[\sum_{i,j}(n_{i|j}(t+1))]} \\ &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t) + \delta_{y_{t+1},i}\delta_{\varphi_{t+1},j}]}{\Gamma[\sum_{i,j}(n_{i|j}(t) + \delta_{y_{t+1},i}\delta_{\varphi_{t+1},j})]} \\ &= \frac{\Gamma[n_{y_{t+1}|\varphi_{t+1}}(t) + 1] \prod_{i\neq y_{t+1}} \prod_{j\neq\varphi_{t+1}} \Gamma[n_{i|j}(t)]}{\Gamma[1 + \sum_{i,j}(n_{i|j}(t))]} \\ &= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t)]}{\Gamma[\sum_{i,j}(n_{i|j}(t))](\sum_{i,j} n(t))} \times \frac{\Gamma[n_{y_{t+1}|\varphi_{t+1}}(t) + 1]}{\Gamma[n_{y_{t+1}|\varphi_{t+1}}(t)]} \\ &= B(n(t)) \times \frac{n_{y_{t+1}|\varphi_{t+1}}(t)}{\sum_{i,j} n_{i|j}(t)}. \end{aligned} \tag{3.15}$$

Combining (3.14) and (3.15) gives the predictive probability in the appealing form of relative frequencies of the observed situations (3.7) modified by the prior guess

$$p(y_{t+1} = i|\varphi_{t+1} = j) = \frac{n_{i|j}(t)}{\sum_{i,j} n_{i|j}(t)} \propto n_{i|j}(t). \tag{3.16}$$

This modified relative frequency can be taken as a point estimate $\hat{\Theta}$ of the unknown parameter Θ. Let us take the expectation of the parameter estimation (3.10). For $k \in y^*$, $r \in \varphi^*$, we have

$$\begin{aligned}
E[p(\theta_{k|r}|D_t)] &= \frac{\int \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)+\delta_{k,i}\delta_{r,j}-1} d\Theta}{B(n(t))} \\
&= \frac{B(n(t)+\delta_{k,i}\delta_{r,j})}{B(n(t))} \\
&= \frac{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t)+\delta_{k,i}\delta_{r,j}]}{\Gamma[\sum_{i,j}(n_{i|j}(t)+\delta_{k,i}\delta_{r,j})]} \times \frac{\Gamma[\sum_{i,j}(n_{i|j}(t)]}{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t)]} \\
&= \frac{\prod_{i\neq k} \prod_{j\neq r} \Gamma[n_{i|j}(t)]\Gamma[n_{k|r}(t)+1]}{\prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \Gamma[n_{i|j}(t)]} \times \frac{\Gamma[\sum_{i,j}(n_{i|j}(t)]}{\Gamma[1+\sum_{i,j} n(i|j)(t)]} \\
&= \frac{\Gamma[n_{k|r}(t)+1]}{\Gamma[n_{k|r}(t)]} \times \frac{1}{\sum_{i,j} n_{i|j}(t)} \\
&= \frac{n_{k|r}(t)}{\sum_{i,j} n_{i|j}(t)}
\end{aligned}$$

Therefore, for MC case, Bayesian prediction is the same as the point estimation of the unknown parameters.

3.3 Parameter Tracking

Adaptivity is the essential feature that allows us to use universal simple models (like ARX) in a range of control and signal processing applications. This ability of reacting to slow changes of the system makes them practically useful. For ARX models, the exponential forgetting is mostly used. Here, we exploit the general Bayesian formulation of this technique [16, 25] in order to obtain an adaptive estimator of Markov chains. Forgetting improves the quality of predictors based on simplified Markov chain models. The ability to follow the changes of a true system is gained.

Let us suppose that the unknown parameters Θ_t are time-variable. The uncertainty of Θ_t given data D_t observed up to time t is described by the posterior pdf $p(\Theta_t|D_t)$. What can be said about the pdf $p(\Theta_{t+1}|D_t)$ of the parameters Θ_{t+1} at the next time instant given the same data D_t?

Suppose that no explicit model of parameter changes is known. Yet, we can quantify our prior information (and possibly information taken from data already available) by introducing an alternative pdf $p^*(\Theta_{t+1}|D_t)$. The problem is then to construct a pdf $p(\Theta_{t+1}|D_t)$ based on two hypotheses:

• parameters remain unchanged.

$$p_1 \equiv p(\Theta_{t+1} = \Theta|D_t) = p(\Theta_t = \Theta|D_t) = \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1} \tag{3.17}$$

• parameters have the worst expected changes.

$$p_2 \equiv p^*(\Theta_{t+1} = \Theta|D_t) = p(\Theta) = \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(0)-1} \tag{3.18}$$

It means we loose all information about Θ_{t+1} but prior information.

We formulate the task of choosing $p(\Theta_{t+1}|D_T)$ given p_1 and p_2 as a Bayesian decision making problem. In particular, we assign probabilities λ and $1-\lambda$ to the hypotheses p_1 and p_2 and define the best estimate $\hat{p}(\Theta)$ of the "true" $p(\Theta_{t+1} = \Theta|D_t)$ as the pdf minimizing expectation of a distance $\mathcal{D}(p \parallel p_i)$ between p and p_i, $i = 1, 2$

$$min_p[\lambda\mathcal{D}(p \parallel p_1) + (1-\lambda)\mathcal{D}(p \parallel p_2)]. \tag{3.19}$$

The distance $\mathcal{D}(\cdot \parallel \cdot)$ is chosen as Kullback-Leibler divergence [26, 27]

$$\mathcal{D}(p \parallel q) = \int p(\Theta) \log \frac{p(\Theta)}{q(\Theta)} d\Theta. \tag{3.20}$$

The above optimization problem is solved by the following theorem [25].

Theorem 1 *Suppose that the pdf p_1 and p_2 are not mutually orthogonal, i.e., their product is not zero almost everywhere. Then the optimation problem (3.19) has a unique solution*

$$\hat{p}(\Theta) \propto p_1^{\lambda} p_2^{1-\lambda}. \tag{3.21}$$

Based on the above theorem, we can obtain the explicit form of $\hat{p}(\Theta)$ for the MC case. Substitution of (3.17) and (3.18) into (3.21) gives

$$\begin{aligned} \hat{p}(\Theta) &\propto \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\lambda(n_{i|j}(t)-1)+(1-\lambda)(n_{i|j}(0)-1)} \\ &\propto \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{\lambda n_{i|j}(t)+(1-\lambda)n_{i|j}(0)-1} \end{aligned} \tag{3.22}$$

Take $\hat{p}(\Theta)$ as the "true" $p(\Theta_{t+1}|D_t)$. The results obtained are summarized as:

Algorithm 1 *MC Estimator and Predictor*

1. *The pdf* $p(\Theta_t = \Theta|D_t)$ *of Markov-chain parameters* Θ *preserves the Dirichlet form*
$$p(\Theta_t = \Theta|D_t) \propto \prod_{i=1}^{m_{y^*}} \prod_{j=1}^{m_{\varphi^*}} \theta_{i|j}^{n_{i|j}(t)-1}.$$
The (sufficient) statistics $n(t)$ *evolves from an optional value* $n(0) > 0$ *(entry-wise) according to the formula*
$$n_{i|j}(t) = (1-\lambda)n_{i|j}(0) + \lambda \left(n_{i|j}(t-1) + \chi\{[y_t, \varphi_t] = [i, j]\}\right)$$
$\chi\{\triangle\}$ *denotes indicator of the set* $\{\triangle\}$. *The optional forgetting factor* $\lambda \in (0, 1]$ *expresses belief assigned to the hypothesis that* Θ *is constant.*

2. *The predictive probability is given by the frequency of the configurations (3.7) for* $[y_{t+1}, \varphi_{t+1}]$
$p(y_{t+1} = i|\varphi_{t+1} = j, D_t) \propto n_{i|j}(t).$

3.4 Selection of Model Order

The above proposed predictor assumes that the model order m is given as a prior information about the system. In this section, we deal with a more general case when a model structure is unknown, i.e., more than one model structure have to be considered as possible. In this case, the predictor (3.16) is conditioned on the uncertain m in the set of the model candidates $\mathcal{M} \equiv \{m_1, m_2, \ldots\}$. Thus, we have a set of model candidates indexed by m

$$p(y_{t+1} = i|\varphi_{t+1} = j_m, D_t, m) \propto n_{i|j_m}(t), \; m \in \mathcal{M}$$

where j_m is a vector in the set of possible values of the regressor φ_t.

Availability of these predictors, parametrized by the model order only, allows us to use Bayesian estimation (3.4) again and compute the posterior pdf's of the competing orders

$$p(m|D_t) \propto \prod_{\tau=1}^{t} p(y_\tau|\varphi_\tau, D_{\tau-1}, m)p(m|D_0) \tag{3.23}$$

where $p(m|D_0)$ is the prior distribution on $\mathcal{M}$. Typically, uniform $p(m|D_0)$ is used and maximum a posteriori probability estimate of the order is chosen.

3.5 Summary

This chapter describes how to theoretically identify MC models for on-line mode via Bayesian approach which is the foundation of the following approximate prediction methodology. The main result is the MC Estimator and Predictor Algorithm 1 in Section 3.3.

Chapter 4

Dimensionality Reduction (I)

The simplicity of Algorithm 1 in Chapter 3 underlines that the dimensionality is the only but serious drawback of MCs. In this chapter, we propose an approximate methodology for MC-based prediction in order to combat the curse of dimensionality. The implementation of this methodology is fully discussed as well.

The layout of this chapter is as follows. Section 4.1 addresses the dimensionality drawback inherent to MCs. The approximate prediction methodology is proposed in Section 4.2. The implementation of this methodology is explained step by step in Sections 4.3, 4.4, 4.5, 4.6. Finally, this chapter is concluded by addressing the pros and cons of this methodology in Section 4.7.

4.1 Curse of Dimensionality

The proposed predictor in Algorithm 1 is conceptually simple as it includes only trivial operations of counting and summing. Dimensionality of this problem prevents, however, to exploit its superior ability to cope with any stationary nonlinear stochastic discretized system.

The dimensionality is reflected in the excessive size $(m_{y^*}m_{u^*})^{m+2}$ of the sufficient statistic $n(t)$. For technically justified values of $m_{y^*}, m_{u^*} \approx 256$ (the discretization exploiting standard converters) and $m \approx 5$ (an experimental upper bound for a large set of dynamic systems), it can hardly be handled by real computers. Even worse, it suffers from lack of real data to shape the sufficient statistic appropriately. Thus, neither the estimation nor the control design can be applied to the problem of this size. This motivates the search for a feasible approximation.

4.2 Approximate Prediction

Although many techniques concerning the aggregation and disaggregation of Markov chains have been proposed in recent years, none of them is suitable for on-line identification and adaptive control. Therefore, we propose the following novel feasible approximate methodology.

The idea behind this methodology is as follows. The "ideal" solution is applicable when the problem dimension is of order 10^2. Thus we have to restrict sizes of input-output spaces to $m_{y^*}, m_{u^*} \leq 3$ and the memory length to $m \leq 2$ before using the above theory. It is achieved by a sort of aggregation of the original states. It limits substantially capabilities of the gained predictors. We see the only promising way how to go out of the dilemma dimensionality versus achievable quality:

1. Several parallel quantizers are applied to discretize the continuous input and output signals into discrete ones with limited amount of state levels in order to downsize the dimensions of input and output state spaces.

2. Several simplified MC models with short regressors are chosen in order to shorten the memory length.

3. These simple MC models are identified on the same data set and the corresponding predictions made.

4. The overall predictor is approximated by mixing the simple predictors together according to Bayesian pooling theory [28].

The structure of this methodology is depicted in Fig.4.1.

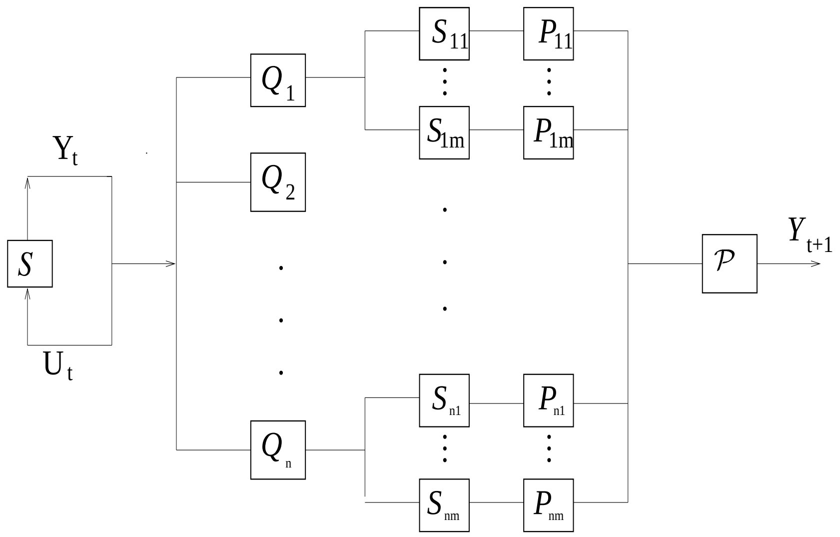

Figure 4.1: Approximate Prediction

where, S is a real system with continuous output Y_t and continuous input U_t; $Q_q, q \in q^* = \{1, \cdots, n\}$ represents quantizers; $S_{qs},\ s = \in s^* = \{1, \cdots, m\}$ is a simple

MC model obtained from the qth quantizer and with sth shortened memory; P_{qs} is the Bayesian predictor gained from the qsth simple MC model; $\mathcal{P}$ is a pooling synthesizer.

In the following sections, we explain designs of quantizers, simple MC models and the pooling synthesizer, respectively.

4.3 Quantizers

The first step of this approximate methodology is a signal discretization, which is realized by several parallel quantizers attached to the system S. The function of quantizers is to decrease the dimensions of the input and output state spaces m_{y^*}, m_{u^*}.

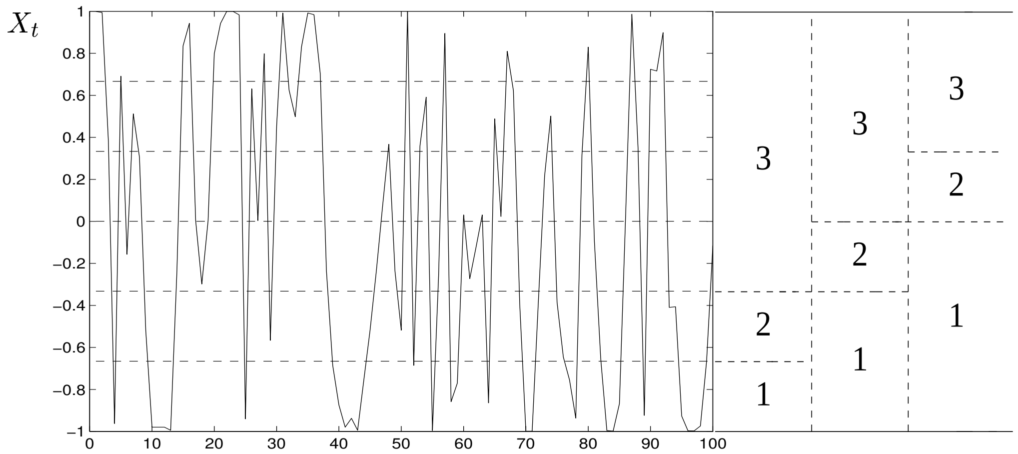

Figure 4.2: Discretization of Continuous Signals

We divide both input and output range into $n+2$ intervals. Each quantizer is determined by the parameter $q = 1, \ldots, n$ that specifies the following three-level quantization function for $X \in \{Y, U\}$, $x \in \{y, u\}$

$$x_t^{[q]} = \begin{cases} 1 & \text{if } \underline{X} \leq X_t < \underline{X}^{[q]} \\ 2 & \text{if } \underline{X}^{[q]} \leq X_t < \overline{X}^{[q]} \\ 3 & \text{if } \overline{X}^{[q]} \leq X_t \leq \overline{X} \end{cases} .$$

where $\underline{X}, \overline{X}$ are (finite) lower and upper bounds of continuous data values, respectively. The interval boundaries $\underline{X}^{[q]}$, $\overline{X}^{[q]}$ are mutually shifted as follows

$$\underline{X}^{[q]} = \underline{X} + \frac{\overline{X} - \underline{X}}{n+2} q, \quad \overline{X}^{[q]} = \underline{X} + \frac{\overline{X} - \underline{X}}{n+2}(q+1).$$

Notice that every three-level quantization has a narrow middle interval 2 and two wide intervals 1 and 3. The narrow interval increase the prediction precision significantly.

After quantization, the original system output Y_t and input U_t are decomposed into n discrete outputs $y_t^{[q]}$ and n discrete inputs $u_t^{[q]}$, $q = 1, \cdots, n$. All possible combinations of different discrete inputs and outputs lead to n^2 discrete subsystems

$$
\begin{aligned}
&p(y_t^{[q_1]}|y_{t-1}^{[q_1]}, \cdots, y_{t-m}^{[q_1]}, u_t^{[q_2]}, \cdots, u_{t-m}^{[q_2]}) \\
&q_1 = 1, \cdots, n, \; q_2 = 1, \cdots, n
\end{aligned}
$$

The dimension of the sufficient statistics for each subsystem is 9^{m+1}. Totally we need to store a $n^2 9^{m+1}$ dimensional collection of statistics. This decreases the original dimension of the sufficient statistics $(n+2)^{2(m+1)}$.

The parallel use of these models is novel. It is important that the refinement of the discretization increases demands on the memory only but not on the amount of the measured data. Just the inevitable information loss caused by the rough discretisation is decreased.

4.4 Memory Length Reduction

Even for the rough discretization above, the dimensionality of each subsystem model still increases exponentially with the memory length m. This fact together with the need to cover the true dynamics of the system have motivated the use of the following simple models indexed by $l = (q_1, q_2, s), \; l \in l^*, l^* = q_1^* \times q_2^* \times s^*$,

$$
\begin{aligned}
&p(y_t^{[l]}|\varphi_t^{[l]}, l) = p(y_t^{[q_1]}|y_{t-s}^{[q_1]}, u_{t-s}^{[q_2]}) \\
&q_1 = 1, \ldots, n, \quad q_2 = 1, \ldots, n, \\
&s = 1, \cdots, m, \quad l = 1, \cdots, n^2 m
\end{aligned} \tag{4.1}
$$

as the basis for approximation.

Adaptive estimation of these models can be done in parallel according to Algorithm 1. They use $n^2 m$ counters $n_{i_l|j_l}(t|l)$ of the situations $[y_\tau^{[l]}, \varphi_\tau^{[l]}] = [i_l, j_l], \; i_l \in \{1,2,3\}; \; j_l \in \{1,2,3\} \times \{1,2,3\}$. Thus, only $27n^2m$-dimensional statistic is needed. Its growth compares favourably to the exponential increase $(n+2)^{2(m+1)}$ of the sufficient statistic dimension.

The models (4.1) were proposed in [29]. We have considered various alternatives too. For instance, the arithmetic mean of s delayed values was used as the regressor, i.e.,

$$
\begin{aligned}
&p(y_t^{[l]}|\varphi_t^{[l]}, l) = p(y_t^{[q_1]}|\frac{1}{s}\sum_{i=1}^{s} y_{t-i}^{[q_1]}, \frac{1}{s}\sum_{i=0}^{s} u_{t-i}^{[q_2]}) \\
&q_1 = 1, \cdots, n, \; q_2 = 1, \cdots, n, \\
&s = 1, \cdots, m, \quad l = 1, \cdots, n^2 m
\end{aligned} \tag{4.2}
$$

No improvements were found comparing to the option (4.1). However, the mean-type regressors are very suitable to a control design. Unlike the present choice, it keeps computation complexity of the control design.

The use of simplified models brings computational complexity to the practicable area as the approximation to the full model adds an acceptable overhead.

4.5 Extension of Probabilities

After quantization and memory length reduction, the original system S is decomposed into n^2m subsystems indexed by $l = 1, \cdots, m_{l^*}$. Each of them has the input $u_t^{[l]}$ and output $y_t^{[l]}$. The dynamics of each subsystem can be described by a simple MC model $p(y_t^{[l]}|\varphi_t^{[l]}, l)$. These subsystems can be easily estimated and the corresponding prediction made according to Algorithm 1 $p(y_{t+1}^{[l]}|\varphi_{t+1}^{[l]}, l)$. However, because of different quantization, these predictions $y_{t+1}^{[l]}$ have different state spaces. In order to make these predictors comparable, we extend them to the original state space, i.e., convert these three-level $y_{t+1}^{[l]}$ to the original $m_{y^*} = n + 2$ levels.

$$
\begin{aligned}
&y_{t+1} = i \;\; \text{if} \;\; Y_{t+1} \in \left(\underline{Y} + \frac{\overline{Y} - \underline{Y}}{n+2}(i-1), \underline{Y} + \frac{\overline{Y} - \underline{Y}}{n+2} i \right) \\
&i = 1, 2, \ldots, n+2.
\end{aligned}
$$

Common probabilities are assigned to individual values of y_{t+1} within the "wide" intervals 1 and 3 of the lth model. It means that for each $l \in l^*$

$$
p_l(y_{t+1} = i|\varphi_{t+1}^{[l]}, l) \propto
$$

$$
\begin{cases}
\frac{p(y_{t+1}^{[l]}=1|\varphi_{t+1}^{[l]}, l)}{q} & \text{if} \quad i < q+1 \\
p(y_{t+1}^{[l]} = 2|\varphi_{t+1}^{[l]}, l) & \text{if} \quad i = q+1 \\
\frac{p(y_{t+1}^{[l]}=3|\varphi_{t+1}^{[l]}, l)}{n+1-q} & \text{if} \quad i > q+1
\end{cases}
$$

4.6 Bayesian Pooling Synthesizer

So far, we have n^2m predictors which describe the common predictive variable y_t. We would like to find the final prediction of y_t based on these predictors. This task is accomplished by Bayesian pooling theory.

Bayesian pooling theory is a method to merge several probabilistic models of the same variable into a single probabilistic description. There are two versions of Bayesian pooling theory.

The first version of Bayesian pooling theory [28, 30] provides the ovarall predictor as a weighted geometric mean of these simple predictors. The referred theory is simple but with many technical details. Thus, just the resulting algorithm is summarized here. The proof of this algorithm can be found in [31].

Algorithm 2 *Bayesian Pooling I*
- *Predictors, labelled by* $l = 1, \ldots, m_{l^*}$,

$$p_l(y_{t+1} = i | \varphi_{t+1}^{[l]} = j_l, l) \propto n_{i|j_l}(t|l)$$

are constructed by Algorithm 1 to m_{l^*} *simple models (4.1).*
- *Auxiliary mixing* (m_{y^*}, m_{y^*}) *matrix* F *is computed with entries*

$$F_{kr} = \sum_{y_{t+1} \in y^*} \ln(p_k(y_{t+1}|\varphi_{t+1}^{[k]}, k)) \ln(p_r(y_{t+1}|\varphi_{t+1}^{[r]}, r)) \; \bar{f}(y_{t+1})$$

$k, r = 1, \ldots, m_{l^*}$. *The weighting pre-prior probability* $\bar{f}$ *of* y_{t+1} *is optional;* $\bar{f}(y_{t+1}) = 1/m_{y^*}$ *is used here.*
- *Mixing* m_{l^*} *vector* v *is calculated as the eigenvector of the matrix* F *corresponding to its maximum eigenvalue. It is computed by the cheap recursion* $v_{new} = Fv_{old}/||Fv_{old}||$ *with* $|| \cdot ||$ *denoting Eucledian norm. The recursion starts with a* $v_{old} > 0$ *and its convergence is guaranteed as the matrix* F *is positive definite and has positive entries. Let* $\bar{v}$ *be such a vector, then the length of the mixing vector is*

$$v = \frac{rank(F)}{\sum_{i=1}^{m_{l^*}} \bar{v}_i} \bar{v}.$$

Note that this length differs from that presented in [28]. It corresponds to the improvement described in [30].
- *The approximation* $\hat{p}(y_{t+1}|\varphi_{t+1})$ *to the ideal predictor* $p(y_{t+1}|\varphi_{t+1})$ *is*

$$\hat{p}(y_{t+1} = i|\varphi_{t+1}) \propto \prod_{l=1}^{m_{l^*}} \left(\frac{n_{i|j_l}(t|l)}{\sum_{y \in y^*} n_{y|j_l}(t|l)} \right)^{v_l}$$

The second version of Bayesian pooling theory [32] is based on information theory. It also provides the compromise made from these simple predictor as a weighted geometric mean. However, the pooling weight v is chosen in a different way. The main idea is as follows.

When using several parametrized models indexed by $l \in l^*$, we need to compare quality of predictors $p_l(y_{t+1}|\varphi_{t+1}^{[l]}, l)$. The Bayes rule gives us possibility to update prior probabilities $v_l(0) \equiv p(l), l \in l^*$ that lth predictor is the best among available predictors to the posterior probabilities $v_l(t) \equiv p(l|D_t)$. Different predictors may be superior at different time instants as we deal with simple models. A forgetting should be used in order to track these changes. The Bayesian forgetting gives the updating rule

$$v_l(t+1) \propto \left[p_l(y_{t+1}|\varphi_{t+1}^{[l]}, l)v_l(t)\right]^{\lambda} [v_l(0)]^{1-\lambda} \tag{4.3}$$

Now we have several competitive predictors $p_l(y_{t+1}|\varphi_{t+1}^{[l]}, l)$ together with belief (probability) $v_l(t)$ that they are the "best" predictors [33]. We would like to "pool" them, to find a best compromise $\hat{p}(y_{t+1}|\varphi_{t+1})$. It is chosen as the minimizer of the expected value of Kullback-Leibler distance (3.20)

$$\sum_{l \in l^*} v_l(t)\mathcal{D}(p \parallel p_l(\cdot|\cdot)).$$

The solution to this optimization problem has the form of the weighted geometric mean

$$\hat{p}(y_{t+1}|\varphi_{t+1}) \propto \prod_{l \in l^*} [p_l(y_{t+1}|\varphi_{t+1}^{[l]}, l)]^{v_l(t)} \tag{4.4}$$

Take $\hat{p}(y_{t+1}|\varphi_{t+1})$ as the true $p(y_{t+1}|\varphi_{t+1})$. The results obtained are summarized as:

Algorithm 3 *Bayesian Pooling II*

1. *Predictors, labelled by* $l = 1, \cdots, m_{l^*}$,

$$p_l(y_{t+1} = i|\varphi_{t+1}^{[l]}, l), \qquad i = 1, \cdots, n+2$$

are constructed by Algorithm 1 applied to mn^2 *simple models (4.1).*

2. *A mixing matrix* v *is updated after measuring and discretising* Y_{t+1}, U_{t+1}. *It has entries*

$$v_l(t+1) \propto \left[p_l(y_{t+1}|\varphi_{t+1}^{[l]}, l)v_l(t)\right]^{\lambda} [v_l(0)]^{1-\lambda}$$

3. *The overall predictor* $\hat{p}(y_{t+1}|\varphi_{t+1})$ *is*

$$\hat{p}(y_{t+1} = i|\varphi_{t+1}) \propto \prod_{l=1}^{m_{l^*}} (p_l(y_{t+1} = i|\varphi_{t+1}^{[l]}, l))^{v_l(t)}.$$

The geometric explanation of Bayesian pooling theory is illustrated in the following figures.

Suppose we have two parametrized simple models which describe the "true" system. The two circles represents two parametrized model sets. Bayesian estimation enables us to choose the best simple model within the set which is closest to the "true" system [33]. The pooled model is chosen as a convex combination of the two

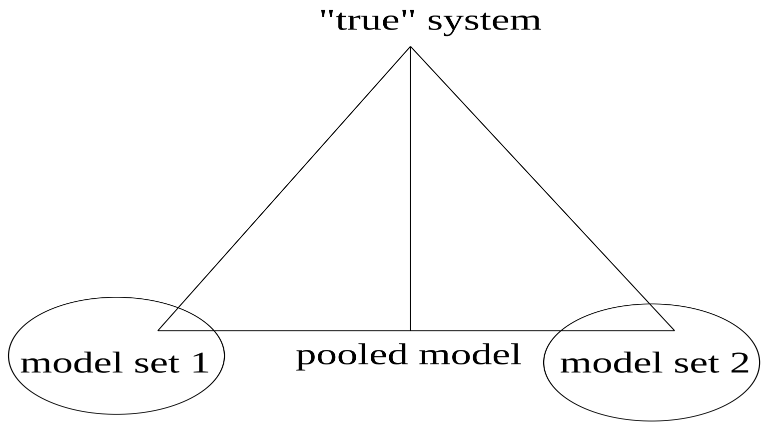

Figure 4.3: Bayesian Pooling Theory

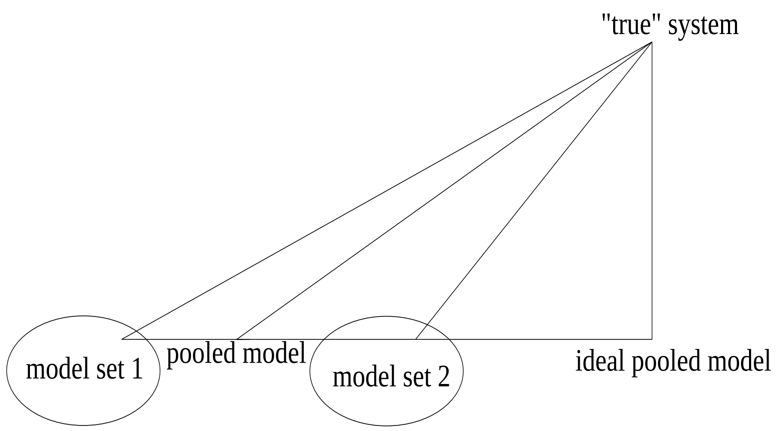

Figure 4.4: Bayesian Pooling Theory

simple models. If the situation in Fig.4.3 happens, the pooled model is better than any of the simple models because it is closest to the "true" system. If the situation in Fig.4.4 happens, the pooled model is worse than one of the simple models. In this case, the pooled model should be chosen as a linear combination of the simple models so that we can find the ideal pooled model. Bayesian pooling can not detect the ideal pooled model in this case so far. The pooling weight v reflects the distance between the simple model and the "true" system.

The two versions of pooling theory have different features. The first version is suitable to the case that we don't have much prior information about the "true" system. The second version works better when we have some prior information about the "true" system. Therefore, the best way to apply the pooling theory is to combine these two versions. It means to use the first version at the beginning of prediction; then swap to the second version.

4.7 Summary

This chapter develops an approximate prediction methodology in order to combat the curse of dimensionality successfully. The dimension of this approximation methodology is $27n^2m$. This is in sharp contrast with the corresponding full Markov predictor whose complexity is $(n+2)^{2(m+1)}$.

Furthermore, this approximation methodology is tested by various experiments which will be covered in the following chapter.

Chapter 5

Prediction Experiments

The approximate prediction methodology is theoretically proposed in the previous chapter. In this chapter we present five experiments in order to indicate the performance of this methodology in both simulated and real systems.

1. The approximate prediction methodology is tested on a particular simulated system in order to compare the performance of the pooled model, simple models and the original model that plays the role of the true system.

2. Monte-Carlo experiment is performed on 500 systems selected at random in order to further confirm the conclusion made in the first experiment.

3. The approximate prediction methodology is tested on a coupled tanks system with nonlinear measurements.

4. Two types of overall predictors are compared on the same simulated system: one uses nonlinear neural network model; the other uses Bayesian pooling theory.

5. Two versions of Bayesian pooling theory are compared both on one particular simulated system and 500 random-selected systems.

The layout of this chapter is in the order of the above five experiments. The conclusion is given in Section 5.6.

5.1 Pooled Model vs. Original Model

This experiment aims at comparing the performance of the pooled model, simple models and original model on one specific simulated system.

5.1.1 Simulated System

The particular simulated system is given by the binomial transition probability

$$p(y_t = i|\varphi_{t-1} = j, \Theta) = \theta_{i|j} = \left(\begin{array}{c} m_{y^*} - 1 \\ i - 1 \end{array} \right) \mu_j^{i-1}(1 - \mu_j)^{m_{y^*} - i}. \tag{5.1}$$

The parameter $\mu_j \in (0, 1)$ determining this distribution is chosen in the form

$$\mu_j \equiv \frac{j}{m_{y^*}^m + 1}.$$

The output has three possible values $m_{y^*} = 3$. The regressor $\varphi_{t-1} = (y_{t-1}, \ldots, y_{t-m})$ with the memory length $m = 3$ is coded to the scalar integer j,

$$j = 1 + \sum_{k=1}^{m} m_{y^*}^{k-1}(y_{t-k} - 1), \; j = 1, \ldots, m_{y^*}^m = 27.$$

Values of $\theta_{i|j}$ are shown in Fig.5.1 and realization of the output in Fig.5.2 .

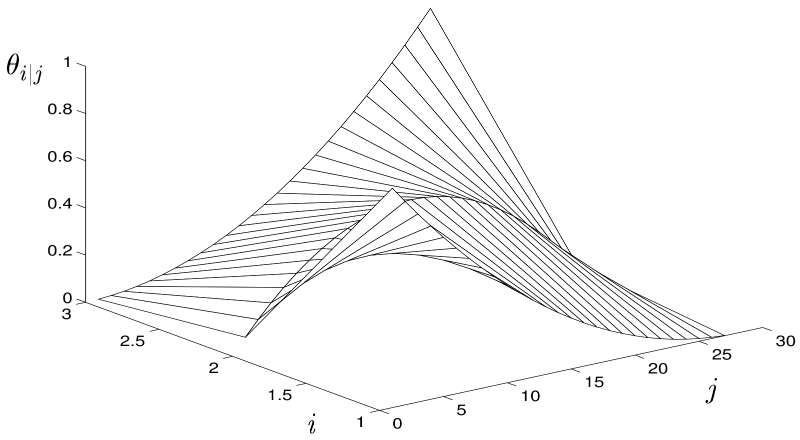

Figure 5.1: Values of Θ

5.1.2 Conditions and Criterion

- three simple models are used

$$p(y_t|y_{t-1}), \; p(y_t|y_{t-2}), \; p(y_t|y_{t-3}).$$

- Uniform prior pdfs on estimated parameters are used ($n(0) = 1$).
- The tested forgetting factors are in the range $\lambda \in [0.91, 1]$.

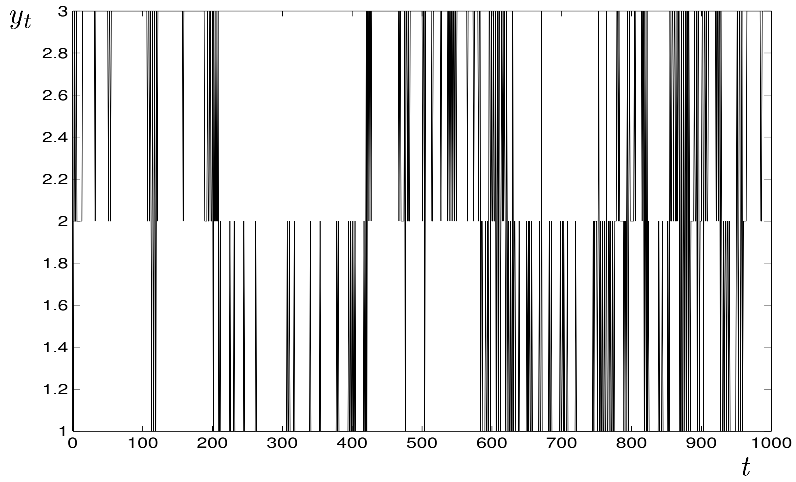

Figure 5.2: Output Sequence

- Quality of any predictor $p(y_{t+1}|\cdot)$ is measured by the sample standard deviation V of the related prediction error

$$\hat{e}_{t+1|t} \equiv y_{t+1} - \sum_{y_{t+1} \in y*} y_{t+1} p(y_{t+1}|\cdot)$$

$$V = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \hat{e}_{t|t-1}}$$

  Here, the expectation assigned to a predictor is taken as the point prediction. T is a simulation length.

- The second version of pooling theory is used.

5.1.3 Results

The results corresponding to 1000 measurements are presented in Fig.5.3. The following correspondence of lines are used

- full line $\quad -\quad$ full model
- dashed line $\quad --\quad$ simple models
- dot-dashed line $\quad \cdot - \cdot\quad$ pooled model

Fig.5.3 shows that all three dashed lines are above the full line and the dot-dashed line for almost all forgetting factors. It means the full model and the pooled model have smaller deviations than all simple models, i.e., the full model and the pooled one provides better predictions than all simple models. Notice that one of the dashed line is very close to the full line. It says that one of the simple models is dominant. It makes much better prediction than others. The intersection of the dot-dashed line and the full line shows that the pooled model is comparable with

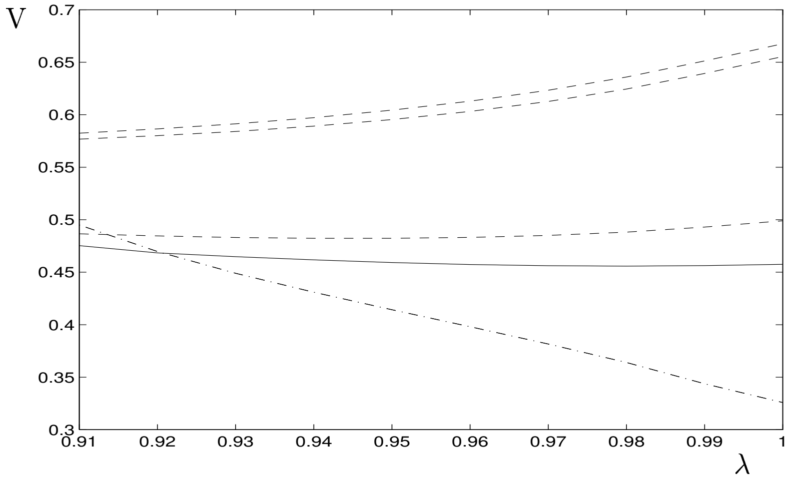

Figure 5.3: Simple x pooled x full predictor

the full model. For bigger forgetting factors, the pooled model is even better than the full one.

5.2 Monte-Carlo Experiment

A Monte-Carlo experiment is performed on 500 systems selected at random in order to further judge the conclusion made in the first experiment.

5.2.1 Simulated Systems

500 systems are generated randomly. The full model of these systems is

$$p(y_t = i|\varphi_{t-1} = j, \Theta) = \theta_{i|j}$$

The regressor is $\varphi_{t-1} = (y_{t-1}, y_{t-2}, y_{t-3})$, i.e., the memory length is $m = 3$. The dimension of the output state space is $m_{y^*} = 3$. The entries of the transition probability matrix $\theta_{i|j}$ are drawn independently from the uniform distribution on (0,1) and normalized to unit sum.

5.2.2 Conditions and Criterion

- For each system, three simple models are used

$$p(y_t|y_{t-1}),\ p(y_t|y_{t-2}),\ p(y_t|y_{t-3}).$$

- The forgetting factor λ is fixed as 0.98.

- The second version of Bayesian pooling theory is applied.

- Quality of 500 sets of predictive results is measured by the histograms of the ratio between deviation of the pooled predictor and deviation of the full predictor $r_P = \frac{V(\text{pooled\_predictor})}{V(\text{full\_predictor})}$ and the ratio between deviation of the best simple predictor and deviation of the pooled predictor $r_S = \frac{V(\text{best\_simple\_predictor})}{V(\text{pooled\_predictor})}$.

5.2.3 Results

The results of this Monte-Carlo experiment are presented in the following figures.

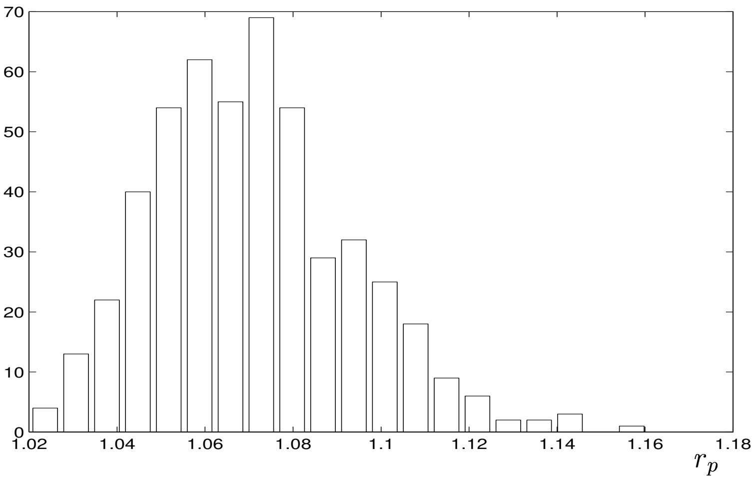

Figure 5.4: Histogram $r_p = \frac{V(\text{pooled\_predictor})}{V(\text{full\_predictor})}$

In Fig.5.4, all values of r_p are more than one. It means the pooled predictor is worse than the full one for all 500 systems because it has bigger deviation than the full predictor. However, majority of r_p are around 1.07, very close to one. It means that the pooled predictor has the similiar predictive result as the full one for majority of 500 systems. It confirms that the results obtained for the particular simulated system in Experiment 1 are not subjectively biased: the same closeness of the full and pooled results for majority of 500 examples is obvious. The cases found when pooled predictors are even better indicate the influence of randomness in the results obtained.

Fig.5.5 shows that the pooled predictor is better than the best simple predictor for 499 systems. It happened only once that the best simple predictor is better than the pooled one. It means the approximation methodology works as desirable in almost all cases.

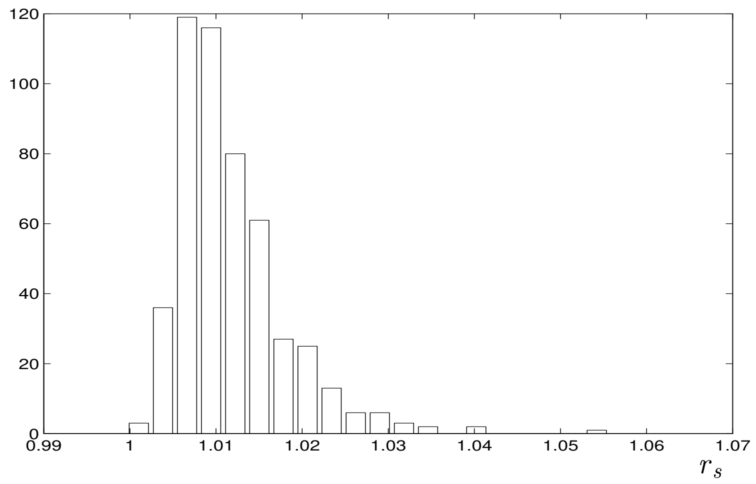

Figure 5.5: Histogram $r_s = \frac{V(\text{best\_simple\_predictor})}{V(\text{pooled\_predictor})}$

5.3 A Coupled Tanks System

In this experiment, we test the proposed approximate prediction methodology on a coupled tanks system with nonlinear measurements in order to check the performance of this approximation methodology on real nonlinear systems.

5.3.1 System Description

The *coupled tanks system* (Fig.5.6) is perhaps one of the most spread physical devices present in the control group laboratories throughout the world.

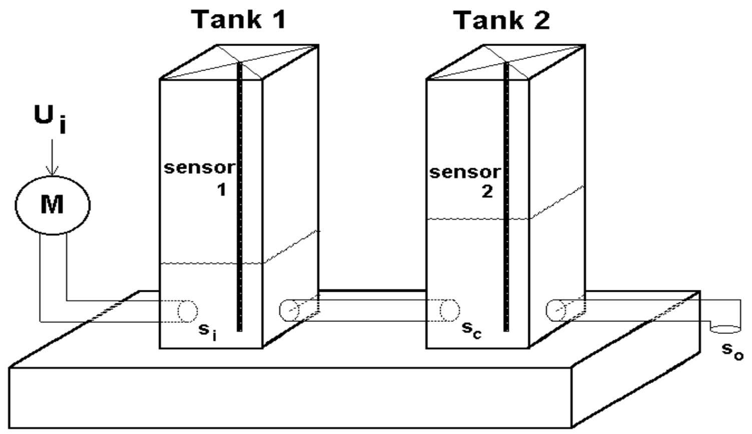

Figure 5.6: The coupled tanks system

The motor M provides a water inflow into the Tank 1. The two tanks are coupled by the tube of the section S_c, the Tank 2 has an outlet tube of the section S_o.

5.3.2 System Modification

A linear regression is used to model this system. It shows the coupled tanks system is practically a linear system. In order to check the performance of the proposed approximate prediction methodology on nonlinear systems, two types of nonlinear elements are attached to the output of the system. One of them is sin function with the frequency 2. Another one is also a sin function but with the frequency 5, i.e.,

$$y_t^1 = sin(2y_t),\ y_t^2 = sin(5y_t).$$

The input remains the same. Therefore, two nonlinear systems are obtained.

5.3.3 Implementation

We suppose that the memory length of these two systems is $m \leq 4$. Four quantizers are used to discretize the continuous input and output signals into 4 three-level discrete signals. Based on discrete signals, $4m$ simple models are identified with exponential forgetting $\lambda = 0.98$ and correspoding predictions are made. These discrete predictions are pooled into the discrete overall predictors by using the first version of Bayesian pooling theory. The discrete overall predictor is transferred into the continuous one by calculating the mean values of the discrete predictive pdf.

5.3.4 Model Order Selection

The model set is chosen as a union of all possible simple models and pooled models so that we can compare the predictive results gained from these two types of models as well as determine the optimal model order. The uniform prior distribution is used, i.e., $f(m) = \frac{1}{M}$ where M is the total number of all possible models. The result says the best model for System One is the $2nd$ order pooled model, while the best model for System Two is the $1st$ order pooled model.

5.3.5 Results

The prediction results corresponding to 100 measurements are presented in the following figures. The following correspondence of lines are used.

- full line $-$ original continuous output
- dashed line $--$ predictive values

Figure 5.7 and 5.8 show that the time courses of the system outputs and their predictions are very close to each other. It means the proposed approximation methodology works properly on real nonlinear systems.

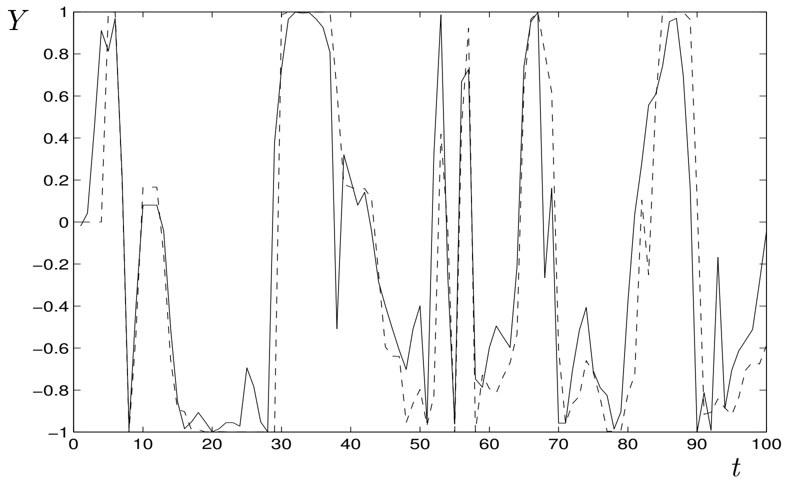

Figure 5.7: The First System

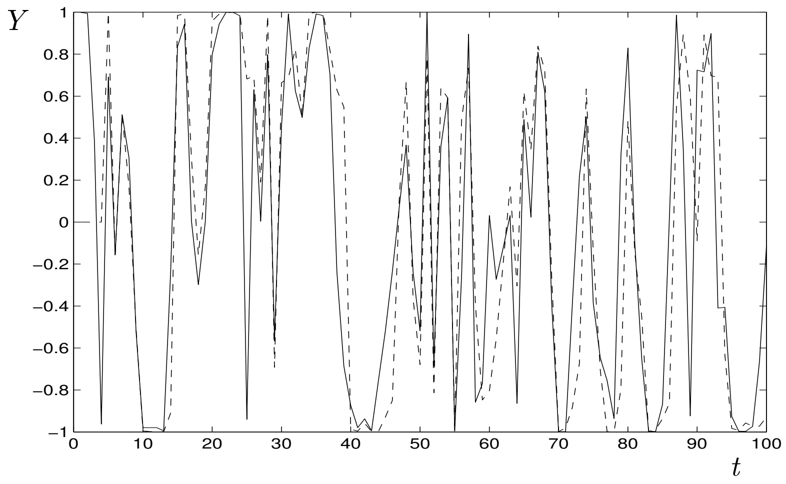

Figure 5.8: The Second System

5.4 MC vs. NN Predictors

Neural Network (NN) as an effective description of complex mapping has been successfully applied to nonlinear system identification/prediction. The pooling theory is also demonstrated to be a powerful tool for approximating high-order MC model [34]. This experiment aims at comparing the performance of NN and pooling theory based predictors.

5.4.1 Simulated System

The same particular autonomous system as in Experiment 1 is given by the binomial transition probability (5.1).

5.4.2 Pooling Theory Predictor

Three simple models are combined

$$p(y_t|y_{t-1}),\ p(y_t|y_{t-2}),\ p(y_t|y_{t-3}). \tag{5.2}$$

Uniform prior pdfs on estimated parameters are used ($n(0) = 1$). The forgetting factor λ is 0.99. The resulting sample mean square prediction error on 1000 samples is 0.5197.

5.4.3 Neural Network Predictor

The desired neural network predictor attempts to find a nonlinear function $\mathcal{F}$ which provides $y_t = \mathcal{F}(y_{t-1}, y_{t-2}, y_{t-3})$ [35].

• **Network Design**

First, we use a network without hidden layer, i.e., single neuron (Fig.5.9).

It is equivalent to a simple nonlinear regression between three inputs and one output. Transfer function is tan-sigmoid. In order to make approach closest to adaptive MC predictor, we use adaptive algorithm Eleane 3 from [36]. This algorithm adapts weights of net on-line. Each coming data is used for adapting of model parameters. We use 900 data points. The first 100 data points were used for stabilization of model parameters. It is necessary due to random initialization of model parameters in the interval [-0.5, 0.5]. We repeated model evaluation for different initial conditions in order to verify that the result does not depend on initial parameter values. Data set 101-1000 was used for performance evaluation of NN adaptive model.

Another network with single hidden layer (Fig.5.10) was also tested. The transfer functions in the hidden layer f are tan-sigmoids. The linear transfer function g is used on the output layer. The size of the NN is determined by the value of the generalization error and by the preference for parsimonious description.

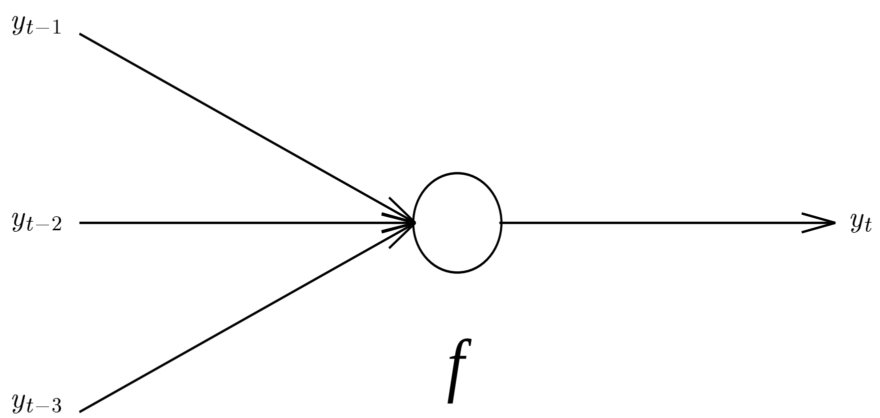

Figure 5.9: Architecture of Network One

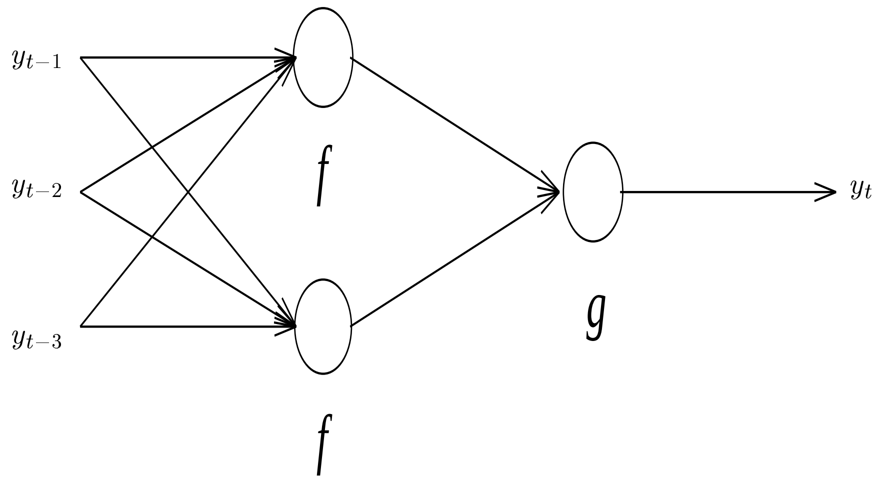

Figure 5.10: Architecture of Network Two

The generalization errors for network One and Two are almost the same. It seems the larger size of the neural network doesn't improve the predictive results. So we prefer to use the simpler network. Network One is chosen as the final neural network predictor.

• **Network Prediction**

We use the selected particular network One to predict the output y_t given three past outputs y_{t-1}, y_{t-2} and y_{t-3}. The mean square prediction error is 0.5599.

5.4.4 Discussion

The presented example demonstrates that the performance of the MC predictor is slightly better than that of NN. Thus we offer a potentially useful complement to the contemporary NN. Because of the long tradition behind the adopted approach there are ready tools which are still weakly supported in NN framework. For instance, it is relatively simple

– to include prior information on the modelled process,
– to estimate structure of MC "network"
– to run it regularly in on line (recursive) mode
– to track parameter changes,
– to consider control applications, etc.

5.5 Bayesian Pooling Theory

In Experiment 1 and Experiment 2, we use Bayesian Pooling Algorithm II. In this experiment, we apply Bayesian Pooling Algirithm I to the same systems in order to compare the performance of these two pooling algorithms. All conditions and criteria in this experiment are kept consistant with Experiment 1 and Experiment 2. The only difference is the different pooling algorithm used. The results are as follows.

5.5.1 Results of One Single System

We apply Bayesian Pooling Algorithm I to the particular system described in Section 5.1.1. The result is presented in Fig. 5.11

Fig.5.11 shows that two of the simple predictors are much worse than the pooled one; while one of them is even better than the pooled one. It means the situation illustrated in Fig. 4.4 happens. The comparison between Fig. 5.3 and Fig. 5.11 tells that the Pooling Algorithm II works better than the Pooling Algorithm I for this system.

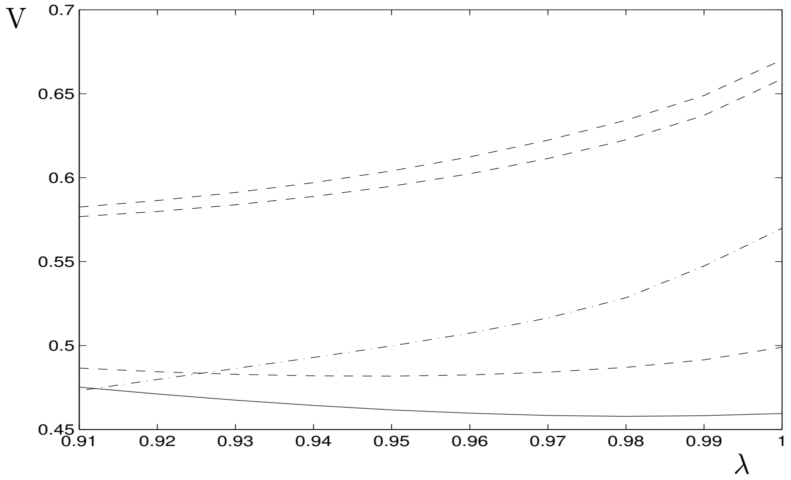

Figure 5.11: Simple x pooled x full predictor

5.5.2 Results of Monto-Carlo Experiment

When we apply Bayesian Pooling Algorithm I to 500 random generated systems described in Section 5.2.1, the following results are obtained.

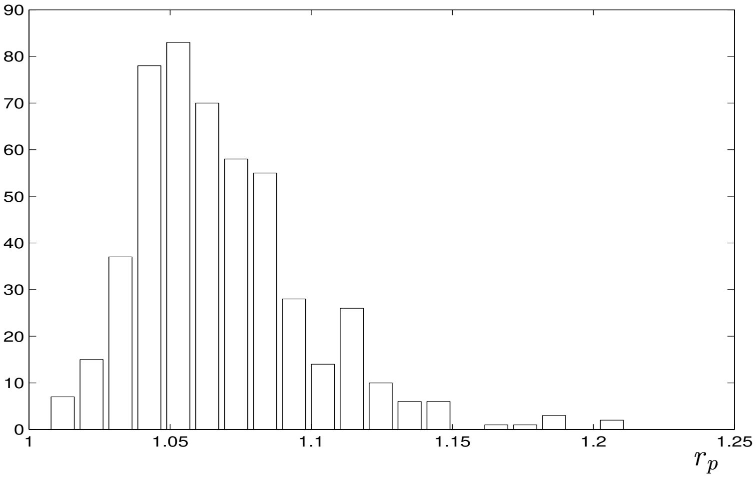

Figure 5.12: Histogram $r_p = \frac{V(\text{pooled\_predictor})}{V(\text{full\_predictor})}$

Fig.5.12 shows that the pooled predictor based on the Pooling Algorithm I is very close to the full predictor. The comparison of Fig.5.12 and Fig. 5.4 says that both pooling algoritms work properly.

Fig.5.13 shows that the situation in Fig. 4.4 doesn't happen very often. Of the 500 random-selected systems, it only happens 71 times. The comparison of Fig.5.13 and Fig. 5.5 tells that the Pooling Algorithm I works a bit worse than the Pooling

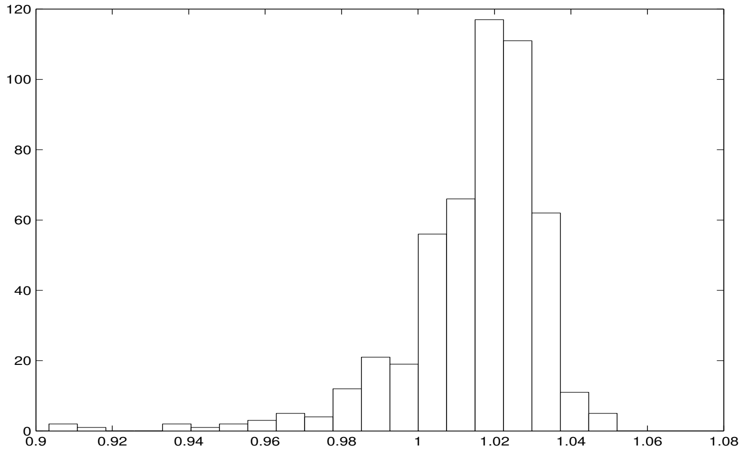

Figure 5.13: Histogram $r_s = \frac{V(\text{best\_simple\_predictor})}{V(\text{pooled\_predictor})}$

Algorithm II. Futher study is needed to improve the pooling theory.

5.6 Summary

In this chapter, the approximate prediction methodology proposed in the previous chapter is tested on five experiments. The first two experiments confirm that this methodology is feasible for simulated systems. The third experiment tells us that this methodology is also applicable to real nonlinear systems. The forth experiment shows that this methodology has almost the same prediction ability with neural network. The last experiment compares the two pooling algorithms. All the experiments in this chapter confirm the feasibility of the proposed methodology.

Chapter 6

Bayesian Adaptive Control

In the previous chapters, we addressed identification problems of unknown systems. Starting from this chapter, we discuss adaptive control problems of unknown systems in which the tasks of identification and control are integrated together. In this chapter we generally formulate the Bayesian adaptive control problem and present one simple example to illustrate complexity and basic features of the Bayesian adaptive control problem.

The layout of this chapter is as follows. Section 6.1 describes the idea of dual control. Section 6.2 introduces two main techniques to reduce the complexity of the adaptive control problem. Section 6.3 summarizes the conventional Bayesian adaptive control design. An example is given in Section 6.4 to illustrate the traditional Bayesian control design. The conclusions are made in Section 6.5.

6.1 Dual Control

Feldbaum's seminal work [37, 38] gave the foundation for the dual control problem. The main idea is that in controlling an unknown system it is necessary to let the controller have dual goals. First, the controller must control the process as well as possible. Second, the controller must inject a probing signal to get more information about the unknown system. By gaining more information better control can be achieved in future steps. The main problem is that to get a better model it is necessary to disturb the process, which is in conflict with the goal of the control purpose. The compromise leads to a controller with dual features.

6.2 Receding Horizon and Certainty Equivalence Strategy

Receding horizon and certainty equivalence strategies are the two most important techniques used for reducing the complexity of the adaptive control problem [39].

In control design, it is better to consider the control horizon H as long as possible. However, this brings computational complexity and excessive computational time, which are not practical for on-line control. The solution is to design controller for a short horizon, then a new controller is calculated when a new measurement is obtained. Since we restart the control design at the beginning of each control period, the control horizon permanently recedes. This method is called *receding horizon strategy*.

Another problem in control design is that the optimal controller depends on not only the random behavior of the system but also the uncertainty of the unknown parameter. It often increases complexity of the control design so that we cannot take into account uncertainty of the unknown parameter in control design. For this reason, the parameter estimate is used as if it is a true one. The uncertainty of the unknown parameter is neglected. This method is called *certainty equivalence strategy*.

Both teachniques are used in the approximate control methodology which will be proposed in the following chapters.

6.3 Bayesian Adaptive Control Design

In this section we present the traditional Bayesian adaptive control design. Receding horizon strategy is considered here. We assume the control horizon is H. The Bayesian approach to adaptive control is this. There is a stochastic dynamic system which depends on a parameter Θ. We are given a prior distribution $p(\Theta)$ for the unknown parameter. At each time instant $t = 1, 2, \cdots$, we obtain a noisy observation y_t of the system. Our goal is to minimize the expectation E of some given loss criterion $L(D_H)$, $E[L(D_H)]$. The control input u_t we apply to the system is on the basis of the data D_{t-1} made on the system, i.e., $R : D_{t-1} \to u_t$. The heart of Bayesian view to this problem is that the expectation in the loss criterion is taken not only with respect to the random behavior of the stochastic system, but also with respect to the random choice of Θ according to $p(\Theta)$.

The standard approach to solving this problem is to transform the problem into an equivalent *dynamic programming* problem [40].

$$
\begin{aligned}
&\min_{R:\{D_{t-1}\to u_t\}_{t=1}^{H}} E[L] = \\
&E\left\{\min_{u_1} E[\ldots \min_{u_{H-1}} E[\min_{u_H} E[L|D_{H-1}, u_H]|D_{H-2}, u_{H-1}] \ldots |D_0, u_1]\right\} \quad (6.1)
\end{aligned}
$$

where R is the causal possibly randomized mapping realized by the controller.

Dynamic programming transfers the problem of finding an optimal mapping $R : D_{t-1} \to u_t$ into the problem of finding the optimal individual input u_t over a finite set u^* (for the considered MC).

This adaptive controller is conceptually simple as it includes just trivial operation of minimization over a finite set. However, dimensionality of this problem prevents to exploit its superior ability to cope with any stationary nonlinear stochastic discretized system. The following example illustrates the complexity of Bayesian adaptive control problem.

6.4 Controlled Tossing of a Coin

In this section, we apply dynamic programming method to control design of a controlled tossing of a coin. It demonstrates complexity and basic features of dual control on a simple stochastic system.

6.4.1 Control Problem

1. **System Model**
   The dynamics of the controlled tossing of a coin system can be described by pdf $p(y_t|D_{t-1}, u_t)$ where t is time instant. The output $y_t \in \{head, tail\} = \{h, a\}$ is observed. The controlled inputs $u_t \in \{head, tail\} = \{h, a\}$ can be interpreted as the original position of the coin and $D_t = \{y_t, y_{t-1}, \cdots, y_1, u_t, u_{t-1}, \cdots, u_1\}$. We further assume that the probability of the next output is determined by the input only,i.e., $p(y_t|D_{t-1}, u_t) = p(y_t|u_t)$.

2. **Criterion**
   The quality of the controller is measured by a loss function $L = \sum_{t=1}^{H} \delta_{y_t,a}$ where $H > 1$ is the control horizon and $\delta_{y_t,a} = \begin{cases} 1 & \text{if } y_t = a \\ 0 & \text{if } y_t = h \end{cases}$. It expresses that the control aim is to toss as many heads as possible.

3. **Control Strategy**
   We search for a causal rule $R : D_{t-1} \to u_t$ which minimizes the expected loss $E(L)$, i.e., $u_t \in Arg \min_R E[L]$ for the following two cases:
   • known parameter case, i.e., $p(y_t|u_t)$ is known.
   • unknown parameter case, supposing the control horizon $H = 2$.

6.4.2 Solution for Known Parameter Case

The solution for the known parameter case is based on the stochastic version of dynamic programming (6.1).

1. Optimal u_H^* is minimizing argument in

$$\min_{u_H} \{E[L|D_{H-1}, u_H]\}$$

$$
\begin{aligned}
&= \min_{u_H} \left\{ E[\sum_{t=1}^{H} \delta_{y_t,a} | D_{H-1}, u_H] \right\} \\
&= \min_{u_H} \left\{ \sum_{t=1}^{H-1} \delta_{y_t,a} + E[\delta_{y_H,a} | D_{H-1}, u_H] \right\} \\
&= \sum_{t=1}^{H-1} \delta_{y_t,a} + \min_{u_H} \{E[\delta_{y_H,a} | u_H]\} \\
&= \sum_{t=1}^{H-1} \delta_{y_t,a} + \min_{u_H} \{p(y_H = a | u_H)\} \\
&= \begin{cases} \sum_{t=1}^{H-1} \delta_{y_t,t} + p(y_H = a | u_H = h) \\ \text{if } p(y_H = a|u_H = h) < p(y_H = a|u_H = a) \\ \sum_{t=1}^{H-1} \delta_{y_t,a} + p(y_H = a | u_H = a) \\ \text{if } p(y_H = a|u_H = h) > p(y_H = a|u_H = a) \end{cases}
\end{aligned}
$$

$$
u_H^* = \begin{cases} h & \text{if } p(y_H = a|u_H = h) < p(y_H = a|u_H = a) \\ a & \text{if } p(y_H = a|u_H = h) > p(y_H = a|u_H = a) \\ h \text{ or } a & \text{if } p(y_H = a|u_H = h) = p(y_H = a|u_H = a) \end{cases}
$$

2. Optimal u_{H-1}^* is minimizing argument in

$$
\begin{aligned}
&\min_{u_{H-1}} \left\{ E[\sum_{t=1}^{H-1} \delta_{y_t,a} + p(y_H = a|u_H^*) | D_{H-2}, u_{H-1}] \right\} \\
&= p(y_H = a|u_H^*) + \min_{u_{H-1}} \left\{ E[\sum_{t=1}^{H-1} \delta_{y_t,a} | D_{H-2}, u_{H-1}] \right\}
\end{aligned}
$$

This is the same situation as when optimizing u_H. Therefore,

$$
u_{H-1}^* = \begin{cases} h & \text{if } p(y_{H-1} = a|u_{H-1} = h) < p(y_{H-1} = a|u_{H-1} = a) \\ a & \text{if } p(y_{H-1} = a|u_{H-1} = h) > p(y_{H-1} = a|u_{H-1} = a) \\ \text{h or a} & \text{if } p(y_{H-1} = a|u_{H-1} = h) = p(y_{H-1} = a|u_{H-1} = a) \end{cases}
$$

3. Optimal u_t^* can be easily found as the similar situation appears for $u_{H-2}, \cdots$. Generally, for each time t,

$$
\boxed{u_t^* = \begin{cases} h & \text{if } p(y_t = a|u_t = h) < p(y_t = a|u_t = a) \\ a & \text{if } p(y_t = a|u_t = h) > p(y_t = a|u_t = a) \\ \text{h or a} & \text{if } p(y_t = a|u_t = h) = p(y_t = a|u_t = a) \end{cases}}
$$

6.4.3 Solution for Unknown Parameter Case

1. Optimal u_H^* is minimizing argument in

$$\begin{aligned}
L_H &\equiv \min_{u_H} \{E[L|D_{H-1}, u_H]\} \\
&= \min_{u_H} \left\{ E\left[\sum_{t=1}^{H} \delta_{y_t,a} | D_{H-1}, u_H \right] \right\} \\
&= \sum_{t=1}^{H-1} \delta_{y_t,a} + \min_{u_H} \{E[\delta_{y_H,a}|D_{H-1}, u_H]\} \\
&= \sum_{t=1}^{H-1} \delta_{y_t,a} + \min_{u_H} \{p(y_H = a|D_{H-1}, u_H)\} \\
&= \sum_{t=1}^{H-1} \delta_{y_t,a} + \min_{u_H} \{\mathsf{p}_H(a|D, u)\}
\end{aligned}$$

where $\mathsf{p}_H(y|D,u) = \frac{n_{y|u}(H-1)}{\sum_y n_{y|u}(H-1)}$ and $n_{y|u}(H) = \sum_{t=1}^{H} \delta_{y_t,y}\delta_{u_t,u}$ increased by prior statistic $n(0)$ corresponding to the self-reproducing pdf (3.12).

$$L_H = \begin{cases} \sum_{t=1}^{H-1} \delta_{y_t,a} + \mathsf{p}_H(a|D,h) & \text{if } \mathsf{p}_H(a|D,h) < \mathsf{p}_H(a|D,a) \\ \\ \sum_{t=1}^{H-1} \delta_{y_t,a} + \mathsf{p}_H(a|D,a) & \text{if } \mathsf{p}_H(a|D,h) > \mathsf{p}_H(a|D,a) \end{cases}$$

$$u_H^* = \begin{cases} h & \text{if } \mathsf{p}_H(a|D,h) < \mathsf{p}_H(a|D,a) \\ a & \text{if } \mathsf{p}_H(a|D,h) > \mathsf{p}_H(a|D,a) \\ a \text{ or } h & \text{if } \mathsf{p}_H(a|D,h) = \mathsf{p}_H(a|D,a) \end{cases}$$

2. Optimal u_{H-1}^* is minimizing argument in

$$\begin{aligned}
L_{H-1} &\equiv \min_{u_{H-1}} \left\{ E\left[\sum_{t=1}^{H-1} \delta_{y_t,a} + \mathsf{p}_H(a|D, u_H^*) | D_{H-2}, u_{H-1} \right] \right\} \\
&= \sum_{t=1}^{H-2} \delta_{y_t,a} + \min_{u_{H-1}} \{E[\delta_{y_{H-1},a} + \mathsf{p}_H(a|D, u_H^*)|D_{H-2}, u_{H-1}]\} \\
&= \sum_{t=1}^{H-2} \delta_{y_t,a} + \min_{u_{H-1}} \left\{ \mathsf{p}_{H-1}(a|D,u) + E\left[\frac{n_{a|u_H^*}(H-1)}{\sum_y n_{y|u_H^*}(H-1)} \,\middle|\, D_{H-2}, u_{H-1} \right] \right\}
\end{aligned}$$

Let us evaluate

$$E\left[\frac{n_{a|u_H^*}(H-1)}{\sum_y n_{y|u_H^*}(H-1)} \,\middle|\, D_{H-2}, u_{H-1} \right]$$

$$= E\left[\frac{n_{a|u_H^*}(H-2) + \delta_{y_{H-1},a}\delta_{u_{H-1},u_H^*}}{\sum_y n_{y|u_H^*}(H-2) + \delta_{u_{H-1},u_H^*}} \;\middle|\; D_{H-2}, u_{H-1}\right]$$

$$= \begin{cases} E\left[\frac{n_{a|h}(H-2)+\delta_{y_{H-1},a}\delta_{u_{H-1},h}}{\sum_y n_{y|h}(H-2)+\delta_{u_{H-1},h}} \mid D_{H-2}, u_{H-1}\right] & \text{if } u_H^* = h \\ E\left[\frac{n_{a|a}(H-2)+\delta_{y_{H-1},a}\delta_{u_{H-1},a}}{\sum_y n_{y|a}(H-2)+\delta_{u_{H-1},a}} \mid D_{H-2}, u_{H-1}\right] & \text{if } u_H^* = a \end{cases}$$

$$= \begin{cases} \mathsf{p}_{H-1}(a|D,u)\left[\frac{n_{a|h}(H-2)+\delta_{u_{H-1},h}}{\sum_y n_{y|h}(H-2)+\delta_{u_{H-1},h}}\right] + \mathsf{p}_{H-1}(h|D,u)\left[\frac{n_{a|h}(H-2)}{\sum_y n_{y|h}(H-2)+\delta_{u_{H-1},h}}\right] \\ \text{if } u_H^* = h \\ \mathsf{p}_{H-1}(a|D,u)\left[\frac{n_{a|a}(H-2)+\delta_{u_{H-1},a}}{\sum_y n_{y|a}(H-2)+\delta_{u_{H-1},a}}\right] + \mathsf{p}_{H-1}(h|D,u)\left[\frac{n_{a|a}(H-2)}{\sum_y n_{y|a}(H-2)+\delta_{u_{H-1},a}}\right] \\ \text{if } u_H^* = a \end{cases}$$

Therefore, the function L_{H-1} has four possible values:

1) when $u_H^* = h,\ u_{H-1}^* = a$

$$\begin{aligned} L_{H-1}^{h1} &\equiv \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,a) + \mathsf{p}_{H-1}(a|D,a)\frac{n_{a|h}(H-2)}{\sum_y n_{y|h}(H-2)} \\ &\quad + \mathsf{p}_{H-1}(h|D,a)\frac{n_{a|h}(H-2)}{\sum_y n_{y|h}(H-2)} \\ &= \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,a) + \frac{n_{a|h}(H-2)}{\sum_y n_{y|h}(H-2)} \\ &= \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,a) + \mathsf{p}_{H-1}(a|D,h) \end{aligned}$$

2) when $u_H^* = h,\ u_{H-1}^* = h$

$$\begin{aligned} L_{H-1}^{h2} &\equiv \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,h) + \mathsf{p}_{H-1}(a|D,h)\frac{n_{a|h}(H-2)+1}{\sum_y n_{y|h}(H-2)+1} \\ &\quad + \mathsf{p}_{H-1}(h|D,h)\frac{n_{a|h}(H-2)}{\sum_y n_{y|h}(H-2)+1} \end{aligned}$$

3) when $u_H^* = a,\ u_{H-1}^* = a$

$$
\begin{aligned}
L_{H-1}^{a1} \quad &\equiv \quad \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,a) + \mathsf{p}_{H-1}(a|D,a) \frac{n_{a|a}(H-2)+1}{\sum_y n_{a|a}(H-2)} \\
&\quad + \mathsf{p}_{H-1}(h|D,a) \frac{n_{a|a}(H-2)}{\sum_y n_{y|a}(H-2)+1}
\end{aligned}
$$

4) when $u_H^* = a,\ u_{H-1}^* = h$

$$
\begin{aligned}
L_{H-1}^{a2} \quad &\equiv \quad \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,h) + \mathsf{p}_{H-1}(a|D,h) \frac{n_{a|a}(H-2)}{\sum_y n_{y|a}(H-2)} \\
&\quad + \mathsf{p}_{H-1}(h|D,h) \frac{n_{a|a}(H-2)}{\sum_y n_{y|a}(H-2)} \\
&= \quad \sum_{t=1}^{H-2} \delta_{y_t,a} + \mathsf{p}_{H-1}(a|D,h) + \mathsf{p}_{H-1}(a|D,a)
\end{aligned}
$$

3. Conclusion For an unknown system, if the control horizon is $H = 2$, the optimal controller is

$$
\boxed{
\begin{array}{c}
u_H^* \in Argmin_{u_H \in \{a,h\}} \{\mathsf{p}_H(a|D,u)\} \\[1em]
u_{H-1}^* \in Argmin_{u_{H-1} \in \{a,h\}} \{L_{H-1}^{u_H^* 1}, L_{H-1}^{u_H^* 2}\}
\end{array}
}
$$

Notice that the One step control design coinsides with certainty equivalent design.

6.4.4 Simulations

The simulations are done to illustrate the properties of the controllers discussed above. Receding horizon strategy is applied to one and two step control designs.

First, we apply the above control design to one specific coin system. The transition probability matrix of the coin system is assumed as $\Theta = \begin{bmatrix} 0.4800 & 0.4200 \\ 0.5200 & 0.5800 \end{bmatrix}$. Suppose this transition probability matrix is unknown. The simulation length is 100. For one step control design, the estimated value of Θ is $\begin{bmatrix} 0.9990 & 0.3939 \\ 0.0010 & 0.6061 \end{bmatrix}$. The loss function is equal to 60.002. For two step control design, the estimated value of Θ is $\begin{bmatrix} 0.5333 & 0.3765 \\ 0.4667 & 0.6235 \end{bmatrix}$. The loss function is also equal to 60.002. Even though the losses are the same for the two cases, the estimates of the unknown parameter Θ are different. The two step control design improve the estimation a lot because it helps the system self-learning.

Second, we carried out a Monte-Carlo experiment to test the above control design. 100 coin systems are generated at random. Among them, the two step control design provides smaller loss than the one step control design for 22 systems; the two step control design is worse than one step control design for 16 systems; for 62 systems they are the same. Very similar pattern was found for various realizations of this experiment.

In spite of approximate nature of the receding horizon strategy, results with two step control design was found slightly but uniformly better than those with one step control.

6.4.5 Discussion

This example reflects generally valid conclusions:

1. Incompleteness of knowledge increases the difficulty of the control design.

2. The dynamic programming characterizes formally the ideal solution to adaptive control design of stochastic systems but it fails to be applicable to realistic systems because of dimensionality problem.

3. The number of options to be compared is $m_{u^*}^{H-1}$; for inputs with more values or a longer horizon the dimensionality bursts.

4. Optimal u_H^* for the unknown parameter case coincides with certainty equivalent solution. The "dual" nature of control design manifests itself for horizon $H > 1$ only when the dynamic nature of learning (estimation) may be seen.

5. The value of optimal u_{H-1}^* influences not only the immediate expected loss, but also the future statistics, i.e., the future estimates. This feature justifies the name "dual" to the resulting control.

6.5 Summary

This chapter serves as a general description of conventional Bayesian adaptive control problem which is illustrated by a simple example. The conclusion is that dynamic programming only provides theoretical solution to Bayesian adaptive control problem. It fails to be feasible in practice because of its computational complexity. Therefore, an alternative control design is needed, which will arise in the next chapter.

Chapter 7

Fully Probabilistic Control Design

The traditional dynamic programming control design of stochastic systems is hardly applicable due to its computational complexity. For this reason, we present an alternative design formulation in this chapter, called fully probabilistic control design [17, 41]. It simplifies the design so that a way is open to a truly adaptive dual version of controller. Its idea is to minimize a suitable distance between the actual and the desired probabilistic description of the closed loop behavior.

The layout of this chapter is as follows. The control problem is formulated via Bayesian approach in Section 7.1. The optimal controller is derived in Section 7.2. Finally, this chapter is summarized in Section 7.3.

7.1 Bayesian Formulation

Consider a stochastic discretized system with single input u_t and single output y_t. The dynamics of this system is described by a stationary controlled Markov chain

$$\begin{aligned}
&p(y_t \mid y_{t-1}, u_t) \\
&y_t \in \{1, 2, \ldots, m_{y^*}\}, \quad m_{y^*} < \infty \\
&u_t \in \{1, 2, \ldots, m_{u^*}\}, \quad m_{u^*} < \infty
\end{aligned}$$

The randomized controller to be designed is described by the transition probability

$$c(u_t \mid y_{t-1}) \tag{7.1}$$

Using the chain rule, we obtain the closed loop description given by the conditional joint pdf of the observed data $D_H = (u_1, \ldots, u_H, y_1, \ldots, y_H)$ and by a priori given state y_0.

$$J(D_H|y_0) = \prod_{t=1}^{H} p(y_t \mid y_{t-1}, u_t)c(u_t \mid y_{t-1})$$

A receding horizon $H > 1$ is applied. The controller $c(u_t \mid y_{t-1})$, which is to be designed, influences partially the joint pdf $J(D_H|y_0)$. Usually, this pdf is shaped by selecting a loss function $L(D_H)$ and minimizing its expected value

$$E[L] = \sum_{D_H} L(D_H)J(D_H|y_0).$$

Let $I(D_H|y_0)$ be a user specified ideal conditional joint pdf of the closed loop data. We can assume that we express it as a formal result of the chain rule in the following way

$$I(D_H|y_0) = \prod_{t=1}^{H} P(y_t \mid y_{t-1}, u_t)C(u_t \mid y_{t-1})$$

where $P(y_t \mid y_{t-1}, u_t)$ is the ideal description of the system and $C(u_t \mid y_{t-1})$ is the ideal controller.

The discrepancy between I and J measured by the Kullback-Leibler distance is

$$\mathcal{D}_H = \mathcal{D}(J \,||\, I) = \sum_{D_H} J(D_H|y_0) ln\left(\frac{J(D_H|y_0)}{I(D_H|y_0)}\right)$$

which is non-negative, zero for $J = I$ (almost surely) and infinite if $I \equiv 0$ and $J > 0$ for some D_H.

7.2 Optimization

Now we minimize the Kullback-Leibler distance between the actual closed-loop description and the ideal joint pdf so that the optimal controller makes this joint pdf $J(D_H|y_0)$ as close as possible to the ideal one $I(D_H|y_0)$.

We can write the Kullback-Leibler distance in the additive form.

$$\begin{aligned}
\mathcal{D}_H &= \sum_{D_H} J(D_H|y_0)\left\{\sum_{t=1}^{H}\left[ln\left(\frac{p(y_t \mid y_{t-1}, u_t)}{P(y_t \mid y_{t-1}, u_t)}\right) + ln\left(\frac{c(u_t \mid y_{t-1})}{C(u_t \mid y_{t-1})}\right)\right]\right\} \\
&= \sum_{t=1}^{H}\sum_{D_H}\left[\prod_{\tau=1}^{H} p(y_\tau \mid y_{\tau-1}, u_\tau)c(u_\tau \mid y_{\tau-1})\right] \cdot \\
&\quad \left[ln\left(\frac{p(y_t \mid y_{t-1}, u_t)}{P(y_t \mid y_{t-1}, u_t)}\right) + ln\left(\frac{c(u_t \mid y_{t-1})}{C(u_t \mid y_{t-1})}\right)\right]
\end{aligned}$$

$$= \sum_{t=1}^{H} \sum_{y_t=1}^{m_{y^*}} \sum_{y_{t-1}=1}^{m_{y^*}} \sum_{u_t=1}^{m_{u^*}} p(y_t \mid y_{t-1}, u_t) c(u_t \mid y_{t-1}) p(y_{t-1}) \cdot \left[ln \left(\frac{p(y_t \mid y_{t-1}, u_t)}{P(y_t \mid y_{t-1}, u_t)} \right) + ln \left(\frac{c(u_t \mid y_{t-1})}{C(u_t \mid y_{t-1})} \right) \right] \quad (7.2)$$

where

$$p(y_{t-1}) = \sum_{y_{t-2}=1}^{m_{y^*}} \sum_{u_{t-1}=1}^{m_{u^*}} p(y_{t-1} \mid y_{t-2}, u_{t-1}) c(u_{t-1} \mid y_{t-2}) p(y_{t-2}) \quad (7.3)$$

Starting at probability concentrated on the given initial state y_0, this probability depends on the previous selected controllers and makes the optimization dynamic for $H > 1$.

We denote

$$\alpha(y_{t-1}, u_t) \equiv \sum_{y_t=1}^{m_{y^*}} p(y_t \mid y_{t-1}, u_t) ln \left(\frac{p(y_t \mid y_{t-1}, u_t)}{P(y_t \mid y_{t-1}, u_t)} \right)$$

and we suppose that this quantity is finite for $t \in \{1, \ldots, H\}$.

Now we optimize the Kullback-Leibler distance (7.2) with respect to $c(u_H \mid y_{H-1})$. We need to minimize

$$\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \sum_{u_H=1}^{m_{u^*}} c(u_H \mid y_{H-1}) \left[\alpha(y_{H-1}, u_H) + ln \left(\frac{c(u_H \mid y_{H-1})}{C(u_H \mid y_{H-1})} \right) \right] \quad (7.4)$$

The factor $c(u_H \mid y_{H-1})$ has no influence on $p(y_{H-1})$. That is why we only have to minimize

$$\triangle(c(\bullet \mid y_{H-1})) \equiv \sum_{u_H=1}^{m_{u^*}} c(u_H \mid y_{H-1}) \left[\alpha(y_{H-1}, u_H) + ln \left(\frac{c(u_H \mid y_{H-1})}{C(u_H \mid y_{H-1})} \right) \right] \quad (7.5)$$

under the normalization restriction $\sum_{u_H=1}^{m_{u^*}} c(u_H \mid y_{H-1}) = 1$ that has to be valid. Because the minimized functional is convex and we minimize it on the convex set, every stationary point of this functional is the point where the functional is minimized.

Now we can write that

$$c(u_H \mid y_{H-1}) = c^*(u_H \mid y_{H-1}) + \varepsilon \delta(u_H \mid y_{H-1}) \quad (7.6)$$

where $c^*(u_H \mid y_{H-1})$ is the searched minimum and δ is its variation. We also want the normalization condition to be valid.

We substitute (7.6) into (7.5) and we get the corresponding form of the minimized functional $\triangle(c(\bullet \mid y_{H-1}))$.

$$\begin{aligned}\triangle(c(\bullet \mid y_{H-1})) &= \sum_{u_H=1}^{m_{u^*}} (c^*(u_H \mid y_{H-1}) + \varepsilon\delta(u_H \mid y_{H-1})) \cdot \\ &\left[ln\frac{c^*(u_H \mid y_{H-1}) + \varepsilon\delta(u_H \mid y_{H-1})}{C(u_{H-1} \mid y_{H-1})} + \alpha(u_H, y_{H-1})\right]\end{aligned}$$

We subtract the term $\lambda \sum_{u_H=1}^{m_{u^*}} (c^*(u_H \mid y_{H-1}) + \varepsilon\delta(u_H \mid y_{H-1}))$ in order to get Lagrangian of the task with the Lagrangian factor λ. We set the first variation of this functional to zero for $\varepsilon = 0$. We get the following relation

$$\sum_{u_H=1}^{m_{u^*}} \delta(u_H \mid y_{t-1}) \left[ln\frac{c^*(u_H \mid y_{H-1})}{C(u_H \mid y_{H-1})} + \alpha(u_H, y_{H-1})\right]$$

$$+ \sum_{u_H=1}^{m_{u^*}} \delta(u_H \mid y_{H-1}) - \lambda \sum_{u_H=1}^{m_{u^*}} \delta(u_H \mid y_{H-1})$$

$$= \sum_{u_H=1}^{m_{u^*}} \delta(u_H \mid y_{H-1}) \left[ln\frac{c^*(u_H \mid y_{H-1})}{C(u_H \mid y_{H-1})} + \alpha(u_H, y_{H-1}) + 1 - \lambda\right] = 0$$

As the $\delta(u_H \mid y_{H-1})$ is arbitrary, the integrand has to be equal to 0. Using this relation we get

$$\frac{c^*(u_H \mid y_{H-1})}{C(u_H \mid y_{H-1})} = exp[-\alpha(u_H, y_{H-1}) + \lambda - 1]$$

Using the normalization restriction for the optimal controller we get that the optimal controller

$$c^*(u_H|y_{H-1}) = \gamma^{-1}(y_{H-1}) \cdot C(u_H \mid y_{H-1})exp[-\alpha(u_H, y_{H-1})]$$

where

$$\gamma(y_{H-1}) = \sum_{u_H=1}^{m_{u^*}} C(u_H \mid y_{H-1})exp[-\alpha(u_H, y_{H-1})]$$

We get the attained minimum by substituting the optimal controller into the minimized relation (7.4)

$$\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \sum_{u_H=1}^{m_{u^*}} \frac{C(u_H \mid y_{H-1})exp[-\alpha(y_{H-1}, u_H)]}{\gamma(y_{H-1})}.$$

$$\left[\alpha(y_{H-1}, u_H) + ln\frac{exp[-\alpha(y_{H-1}, u_H)]}{\gamma(y_{H-1})}\right]$$

$$= - \sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \sum_{u_H=1}^{m_{u^*}} \frac{C(u_H \mid y_{H-1}) exp[-\alpha(y_{H-1}, u_H)]}{\gamma(y_{H-1})} ln(\gamma(y_{H-1}))$$

Because $ln(\gamma(y_{H-1}))$ does not depend on u_H, we can rewrite this relation in the following way

$$- \sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) ln(\gamma(y_{H-1})) \sum_{u_H=1}^{m_{u^*}} c^*(u_H \mid y_{H-1})$$

Due to the normalization restriction, the attained minimum

$$\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \triangle(c^*(\bullet \mid y_{H-1})) = - \sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) ln\gamma(y_{H-1}) \tag{7.7}$$

Now we continue the minimization of the Kullback-Leibler distance (7.2) with respect to $c(u_{H-1} \mid y_{H-2})$. It influences the sum of the second last term in the additive expression of (7.2) and $\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \triangle(c^*(\bullet \mid y_{H-1}))$. We can derive the dependence of (7.7) on the controller to be chosen using the equation describing the evolution of the unconditional pdf $p(y_t)$ (7.3).

$$\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) \triangle(c^*(\bullet \mid y_{H-1})) \equiv - \sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1}) ln(\gamma(y_{H-1})) =$$

$$- \sum_{y_{H-2}=1}^{m_{y^*}} p(y_{H-2}) \sum_{u_{H-1}=1}^{m_{u^*}} \left[\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1} \mid y_{H-2}, u_{H-1}) ln(\gamma(y_{H-1}))\right] c(u_{H-1} \mid y_{H-2})$$

Therefore, the factor $c(u_{H-1} \mid y_{H-2})$ influences the following sum

$$- \sum_{y_{H-2}=1}^{m_{y^*}} p(y_{H-2}) \sum_{u_{H-1}=1}^{m_{u^*}} \left[\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1} \mid y_{H-2}, u_{H-1}) ln(\gamma(y_{H-1}))\right] c(u_{H-1} \mid y_{H-2})$$

$$+ \sum_{u_{H-1}=1}^{m_{u^*}} \sum_{y_{H-2}=1}^{m_{y^*}} \sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1} \mid y_{H-2}, u_{H-1}) c(u_{H-1} \mid y_{H-2}) p(y_{H-2}) \cdot$$

$$\left[ln\frac{p(y_{H-1} \mid y_{H-2}, u_{H-1})}{P(Y_{H-1} \mid y_{H-2}, u_{H-1})} + ln\frac{c(u_{H-1} \mid y_{H-2})}{C(u_{H-1} \mid y_{H-2})}\right]$$

Denote

$$\beta(y_{H-2}, u_{H-1}) = -\sum_{y_{H-1}=1}^{m_{y^*}} p(y_{H-1} \mid y_{H-2}, u_{H-1}) ln(\gamma(y_{H-1})) \tag{7.8}$$

We have to minimize the following quantity

$$\begin{aligned} \triangle c(\bullet \mid y_{H-2})) \quad &\equiv \quad \sum_{u_{H-1}=1}^{m_{u^*}} c(u_{H-1} \mid y_{H-2}) \Bigg[\alpha(y_{H-2}, u_{H-1}) + \beta(y_{H-2}, u_{H-1}) \\ &+ \quad ln\left(\frac{c(u_{H-1} \mid y_{H-2})}{C(u_{H-1} \mid y_{H-2})}\right)\Bigg] \end{aligned} \tag{7.9}$$

If we compare this quantity with (7.5), we see that we are in the same situation as there. Just α is replaced by the sum $\alpha + \beta$. It is completely identical if we define $\gamma(y_H) = 1$, i.e., $\beta(y_{H-1}, u_{H-1}) = 0$ defined by (7.8). The analogy of (7.5) and (7.9) leads directly to the optimal controller $c(\bullet \mid \bullet)$ at the time $H - 1$

$$\begin{aligned} c^*(u_{H-1} \mid y_{H-2}) &= \gamma^{-1}(y_{H-2}) C(u_{H-1} \mid y_{H-2}) \cdot \\ &\quad exp[-\alpha(x_{H-2}, u_{H-1}) - \beta(y_{H-2}, u_{H-1})] \\ \gamma(y_{H-2}) &= \sum_{u_{H-1}=1}^{m_{u^*}} C(u_{H-1} \mid y_{H-2}) exp[-\alpha(y_{H-2}, u_{H-1}) - \beta(y_{H-2}, u_{H-1})] \end{aligned}$$

The optimal controller for all $t \in t^*$ can be derived by continuing this "backward" recursion. The result can be summarized in:

Algorithm 4 *The optimal controller $c^*(\bullet \mid \bullet)$ minimizing the Kullback-Leibler distance (7.2) between the actual and the ideal joint pdfs has the form*

$$\begin{aligned} c^*(u_t \mid y_{t-1}) &= \gamma^{-1}(y_{t-1}) C(u_t \mid y_{t-1}) exp[-\alpha(u_t, y_{t-1}) - \beta(u_t, y_{t-1})] \\ \alpha(y_{t-1}, u_t) &= \sum_{y_t=1}^{m_{y^*}} p(y_t \mid y_{t-1}, u_t) ln\left(\frac{p(y_t \mid y_{t-1}, u_t)}{P(y_t \mid y_{t-1}, u_t)}\right) \\ \beta(u_t, y_{t-1}) &= -\sum_{y_t=1}^{m_{y^*}} ln(\gamma(y_t)) p(y_t \mid y_{t-1}, u_t) \\ \gamma(y_{t-1}) &= \sum_{u_t=1}^{m_{u^*}} C(u_t \mid y_{t-1}) exp[-\alpha(u_t, y_{t-1}) - \beta(u_t, y_{t-1})] \\ \gamma(y_H) &= 1 \end{aligned}$$

The only prerequisite to apply this algorithm is the output prediction $p(y_t \mid y_{t-1}, u_t)$, which is addressed in Chapter 3.

7.3 Summary

A new control design methodology called fully probabilistic control design is outlined in this chapter. We only discuss this control design for discrete systems. The continuous version can be found in [17]. Its explicit form (no minimization appears in the final algorithm) is its main advantage.

Based on this design, the approximate prediction methodology proposed in Chapter 4 can be extended to the adaptive control case. The next chapter addresses this extension.

Chapter 8

Dimensionality Reduction (II)

This chapter is devoted to extension of the approximate prediction methodology proposed in Chapter 4 to the adaptive control case.

The layout of this chapter is as follows. The approximate control methodology is proposed in Section 8.1. The implementation of this methodology is described in Sections 8.2, 8.3, 8.4 and 8.5. Lastly, we summarize this chapter in Section 8.6.

8.1 Approximate Control

Fully probabilistic control design Algorithm 4 simplifies the design equations and is more feasible in the implementation of the control design. Based on this control design, we can extend the approximate prediction methodology proposed in Chapter 4 to the on-line control case. The approximate control methodology is like this.

1. Several parallel filters are applied to both continuous valued system input and output signals so that the memory length is shortened to one.

2. For each specific filter, several quantizers are used to discretize the continuous valued signals into a finite amount of discrete ones in order to downsize the dimensions of the state space and control space.

3. After filtering and quantization, the original system is decomposed into a set of simple discrete subsystems and the dynamics of these subsystems can be modelled by the first order CMCs.

4. Probabilistic control design is applied to the first order CMCs and we obtain the optimal subcontrollers for these subsystems.

5. The overall controller is obtained by combining these optimal subcontrollers together using Bayesian pooling theory (4.6).

The structure of this approximate control methodology is depicted in Fig.8.1.

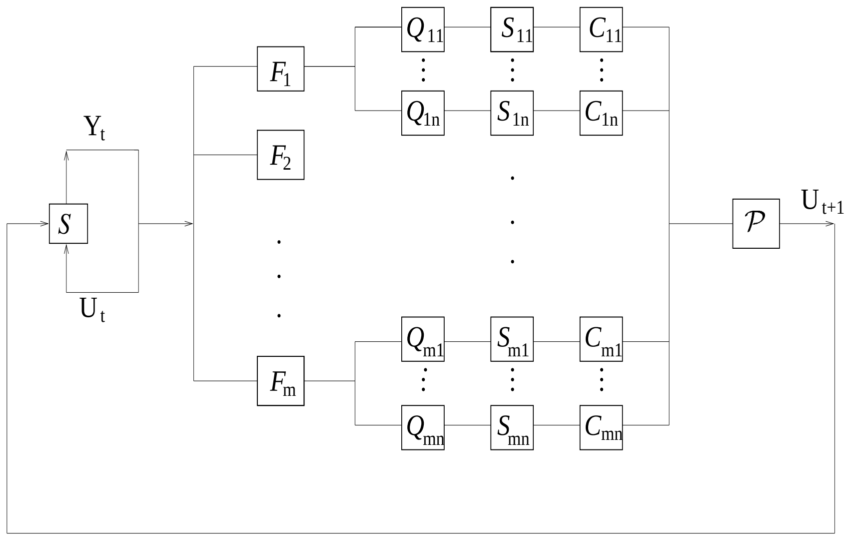

Figure 8.1: Approximate Control

where S is a stochastic system with continuous valued output Y_t and continuous valued input U_t; $F_f, f \in f^* = \{1, \cdots, m\}$ are filters; Q_{fq}, $q \in q^* = \{1, \cdots, n\}$ represents quantizers; S_{fq} are the simple subsystems obtained from the fth filter and qth quantizer; C_{fq} are the optimal subcontrollers; $\mathcal{P}$ is a pooling synthesizer.

In the following sections we explain the designs of filters, quantizers and pooling synthesizer, respectively. The quantizers and pooling synthesizer used in this methodology are similar to those for prediction in Chapter 4. The design of controllers was discussed in Chapter 7. Therefore, the emphasis is put on the explanation of the filter design.

8.2 Parallel Filters

We suppose that the dynamics of the stochastic system S is modelled by a mth order CMC

$$p(Y_t|U_t, Y_{t-1}, U_{t-1}, \cdots, Y_{t-m}, U_{t-m}).$$

The first step of the approximate control of this system is signal filtering, which is realized by several parallel filters attached to the system. The aim of filtering is to shorten the memory length into one.

We use a different shortening-memory way from that for prediction (4.4). In prediction (4.4) we use the following simple models

$$p(y_t|y_{t-s}, u_{t-s}), \qquad s = 1, 2, \cdots, m. \tag{8.1}$$

If we convert (8.1) into a first order CMC, we need to make the transformation $z_t = [y_t, y_{t-1}, \cdots, y_{t-s+1}, u_t, u_{t-1}, \cdots, u_{t-s+1}]'$ so that the new system model is $p(z_t|z_{t-1}, u_t)$, where $[\cdot]'$ means transpose. However, this transformation increases the state space dimension exponentially because the new state space dimension is $(m_{y^*} m_{u^*})^s$, which makes the on-line control design unfeasible.

The new idea to reduce the memory length is to project the original condition $V_t = [Y_{t-1}, \cdots, Y_{t-m}, U_{t-1}, \cdots, U_{t-m}]'$ on several simple uncorrelated signals $Y_t^{[f]}$, $f \in f^* = \{1, \cdots, m\}$. This mapping is realized in the following way:

- Select a new basis $\{w^{[f]}\}_{f \in f^*}$ of the space V^*.
- Expand $V_t = \sum_{f \in f^*} Y_t^{[f]} w^{[f]}$.
- Take $Y_t^{[f]}$ as the fth filtered signal for further modelling and control.

At present state of knowledge, the new basis that makes $Y_t^{[f]}$ mutually uncorrelated seems to be good option as:

- It seperates individual controlled variables.
- It projects the whole vector V_t to each simple model.

Construction of this new basis is straightforward. Let R_t the sample covariance matrix that can be computed recursively using an exponential window, i.e., $R_t = cov(V_t)$. According to Cholesky decomposition, there exists an upper triangle matrix W so that $R_t = WW'$. Take each column of W, $w^{[f]}$, as a new basis of the space V^*. Expanding V_t gives $V_t = WZ$ where $Z = [Y_t^{[1]}, \cdots, Y_t^{[m]}]'$. Therefore, the fth filtered signal $Y_t^{[f]}$ is the fth column of the matrix $W^{-1}V_t$, i.e., $Y_t^{[f]} = (W^{-1}V_t)^{[f]}$. These filtered signals $Y_t^{[f]}$ are mutually uncorrelated because $cov(Z) = cov(W^{-1}V_t) = W^{-1}cov(V_t)(W^{-1})' = W^{-1}R_t(W^{-1})' = I$.

$Y_t^{[f]}$ depends only on $Y_{t-1}^{[f]}$ because V_t depends only on V_{t-1},. We take $Y_t^{[f]}$ as a new subsystem output, and the new input $U_t^{[f]}$ still remain the same as U_t. We use different notations in order to keep the consistancy with the output. Therefore, we obtain a set of the new subsystems which are modelled by the first order CMCs

$$p(Y_t^{[f]}|Y_{t-1}^{[f]}, U_t^{[f]}), \ f = 1, \cdots, m. \tag{8.2}$$

8.3 Parallel Quantizers

Notice that the system input and output after filtering are still continuous valued. Therefore, the second step of the approximation methodology is to discretize the continuous valued signals $Y_t^{[f]}$ and $U_t^{[f]}$ into discrete ones with a limited amount of levels. This function is realized by n parallel quantizers. These quantizers have the same structure with those we used in approximate prediction in Section 4.3. Therefore, we don't intend to go into details about the quantizer design here.

As for each specific filter we apply n quantizers to the filtered output $Y_t^{[f]}$ indexed by q_1 and also n quantizers to the filtered input $U_t^{[f]}$ indexed by q_2, we obtain mn^2

new input-output data pairs:

$$\begin{aligned}&[y_t^{[fq_1]}, u_t^{[fq_2]}]\\&f = 1, 2, \cdots, m,\\&q_1 = 1, 2, \cdots, n,\\&q_2 = 1, 2, \cdots, n\end{aligned}$$

These data pairs are the mn^2 new subsystems' inputs and outputs.

8.4 Controllers

After filtering and quantization, the original system S is decomposed into $L = mn^2$ subsystems indexed by $l = 1, 2, \cdots, L$. Each of them has the input $u_t^{[l]}$ and the output $y_t^{[l]}$. The dynamics of each subsystem S_l can be described by a first order CMC

$$p(y_t^{[l]} | y_{t-1}^{[l]}, u_t^{[l]}) \tag{8.3}$$

These subsystems can be easily estimated according to Algorithm 1 and the corresponding control design made according to Algorithm 4. However, applying the probabilistic control design to each subsystem S_l requires the prior knowledge of the ideal subsystem and ideal subcontroller for each subsystem. In practice, we only know the ideal system and ideal controller of the original system. We can estimate the ideal subsystem and ideal subcontroller as follows:

• ideal input-output data are generated from the original ideal system and the ideal controller.

• ideal data are processed in the same way as real data, i.e., first filtering, then quantization.

• the processed data are used to estimate the ideal subsystems and ideal subcontrollers according to Algorithm 1.

We apply the probabilistic control design for each subsystem and obtain the optimal subcontrollers $p(u_t^{[l]}|\bullet)$.

8.5 Pooling Synthesizer

Because of quantization, these $u_t^{[l]}$ have different state spaces. In order to make these subcontrollers comparable, we extend them to the original space,i.e., convert these three-level $u_t^{[l]}$ to the original $m_{u^*} = n + 2$ levels. The extension $p(u_t^{[el]}|\bullet)$ of the lth subcontroller $p(u_t^{[l]}|\bullet)$ is done in the same way as in Section 4.5.

The overall controller $p(u_t|\bullet)$ is obtained by combining these $p(u_t^{[el]}|\bullet)$ together using Bayesian pooling theory (see Section 4.6). The problem is that we need some kind of measurement to judge the control quality of the individual subcontrollers in order to update the pooling weight. For prediction, we use predictive probability to compare the quality of predictors. In the control case, we also use predictive probability as measurement of control quality as there is no alternative way. Because of filtering, the way we compare the qualities of predictors is different from that for prediction. Here we use $p_l(y_t|y_{t-1}^{[l]}, u_t^{[l]})$. It means we use the filtered data to predict the original system output in order to judge their prediction quanlities. In this case, the updating rule of the pooling weights becomes

$$v_l(t) \propto \left[p_l(y_t|y_{t-1}^{[l]}, u_t^{[l]})v_l(t-1)\right]^{\lambda} [v_l(0)]^{1-\lambda}. \tag{8.4}$$

where λ is the forgetting factor. The overall controller becomes

$$p(u_t|\bullet) \propto \prod_{l \in l^*} [p(u_t^{[el]}|\bullet)]^{v_l(t-1)}. \tag{8.5}$$

Notice that $p(u_t|\bullet)$ is still discrete. The equivalent continuous overall controller $p(U_t|\bullet)$ can be obtained by a random generator.

8.6 Summary

In this chapter, we extend the approximate prediction methodology to the adaptive control case. The implementation of this methodology is described in details. This methodology is based on certainty equivalent strategy. The problem to extend it so as to let the control law be a function of the parameters as well as of the uncertainties is an appealing task for further research.

Moreover, better filters, quantizers and pooling synthesizer still need investigation.

Chapter 9

Control Experiments

The approximate control methodology has been theoretically proposed in the previous chapter. In this chapter we investigate four types of experiments in order to test the performances of filtering, quantization, probabilistic control design and the approximate control methodology, respectively.

1. Two experiments are execucated in order to study the function of quantizers; one is on a continuous system; the other is in a regulation case.
2. The efficiency of filtering is tested on a discrete system with a small state space, but a long memory.
3. Probabilistic control design is tested on a simple discrete system.
4. The approximate control methodology is tested on a coupled tanks system.

Receding horizon and certainty equivalence stretegies are applied to all experiments in this chapter. The layout of this chapter is in the order of the above four experiments. The conclusion is given in Section 9.5.

9.1 Quantizer Tests

In this part we apply quantization to the first order continuous systems. Discretization of continuous signals usually leads to big state spaces. Therefore, quantization is extremely useful for such systems.

9.1.1 Continuous Case

This experiment aims at illustrating quantization on one first order continuous system. We try to control the system output to some set point by using the approximate control methodology. As the memory length of this system is one, it is not necessary to apply any filter. Quantizers play an important role in this experiment. Therefore, the whole approximate control procedure for this experiment is that we apply n quantizers to the original system output and input signal in order to discretize the continuous valued signals into the three-valued discrete ones so that n^2 simple CMCs are obtained; probabilistic control design are made based on these simple CMCs; the final control action is decided by a pooling synthesizer.

• **model**

The dynamics of this continuous stochastic system is modelled by an ARX model of the following structure

$$Y_t = 0.9691Y_{t-1} + 0.0165U_t + 0.0001e_t$$

where $Y_t \in [0, 0.2]$ is the continuous valued system output, $U_t \in [0.2, 0.4]$ is the continuous valued system input, e_t is Gaussian white noise $N(0, 1)$.

• **setting of parameters**

1. quantization number $n = 8$;
2. forgetting factor for estimation $\lambda = 0.95$;
3. control horizon $H = 10$;
4. simulation length $T = 300$;
5. set point $Y_t = 0.15$;
6. forgetting factor for pooling $\lambda = 0.5$.

• **result**

The control aim of this experiment is to make the system output as close to 0.15 as possible.

The result is shown in Fig.9.1.

This figure says that the system output starts to converge to the set point after 100 time instants. It means quantization is able to achieve the expected goal. This idea is feasible in practice.

Similiar experiment is also done for another continuous system $Y_t = 0.98Y_{t-1} + 0.02U_t + 0.00001e_t$. The simulation result is more or less the same.

9.1.2 Regulation Case

This experiment is devoted to test quantizers on the regulation problem of a continuous stochastic system. The approximate control procedure is the same as the above experiment. However, there is a large noice in the system of this experiment. The control aim is to regulate the system output from the big system disturbance.

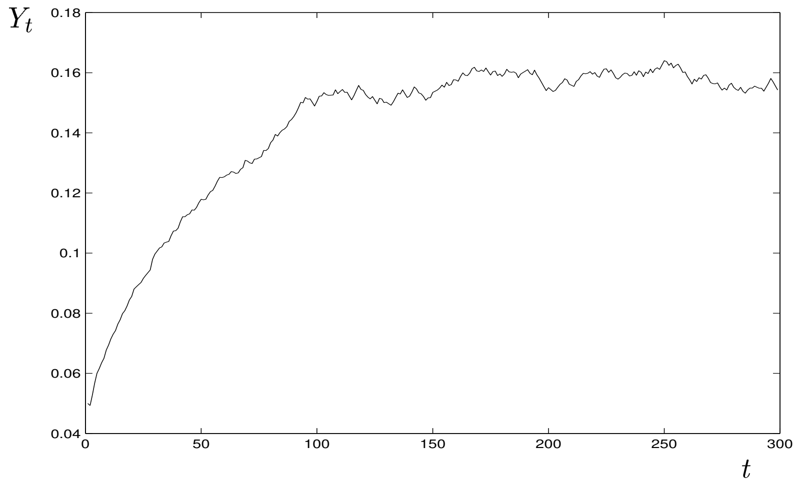

Figure 9.1: System Output

• **model**
The dynamics of this continuous stochastic system is modelled by an ARX model of the following structure

$$Y_t = aY_{t-1} + bU_t + ce_t$$

The coefficients of this model has the following values

$$a = 0.98,\ \ b = 0.02,\ \ c = (1 - a^2)\sigma^2$$

where $\sigma = 0.1$ is the output boundary, $U_t \in [0.2, 0.4]$ is the system input, e_t is Gaussian white noise $N(0, 1)$.
• **setting of parameters**

1. quantization number $n = 18$;
2. forgetting factor for estimation $\lambda = 0.95$;
3. control horizon $H = 10$;
4. simulation length $T = 1000$;
5. set point $Y_t = 0$;
6. forgetting factor for pooling $\lambda = 0.5$.

• **result**
The control aim of this experiment is to make the system output as close to zero as possible.

The result is shown in Fig.9.2.

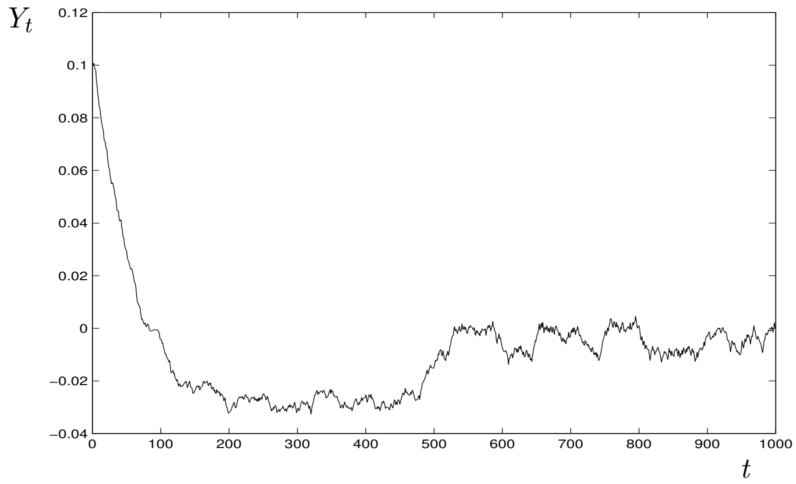

Figure 9.2: System Output

This figure says that the system output starts to converge to the set point after 500 time instants. It means quantization is a feasible way to downsize the dimension of the state space.

9.2 Filter Test

In this experiment, we examine the performance of filters on one discrete system with a small state space but a long memory. The aim of this experiment is to control the system output to some set point by applying the approximete control methodology. As this discrete system has a small state space, no quantizer is needed. Filtering becomes a key point in this approximate control. The whole approximate control procedure is like this: we apply m filters to the system output so that m first order simple CMCs are obtained; probabilistic control design are made based on these simple CMCs; the final controller is obtained by mixing these simple controllers together.

• **model**

A discrete system

$$p(y_t|u_t, y_{t-1}, u_{t-1}, y_{t-2}, u_{t-2}, y_{t-3}, u_{t-3})$$

is generated at random. Both the input and output have three possible values, $m_{y^*} = m_{u^*} = 3$. The memory length of this system is $m = 3$.

• **setting of parameters**

1. filtering number $m = 3$;

2. forgetting factor for estimation $\lambda = 0.95$;

3. control horizon $H = 10$;

4. simulation length $T = 200$;

5. set point $y_t = 3$;

6. forgetting factor for pooling $\lambda = 0.5$.

• **result**
The control aim of this experiment is to make the system output as close to three as possible.

The result is shown in Fig.9.3.

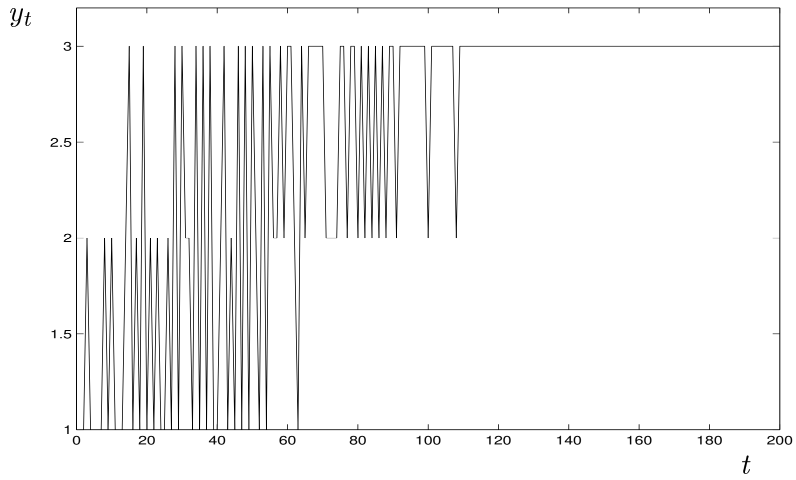

Figure 9.3: System Output

This figure says that the system output starts to converge to the set point after 100 time instants. It means filtering is a reasonable way for the memory reduction.

9.3 Control Design

In this experiment, we intend to test the performance of the proposed control design method, fully probabilistic control design on a simple discrete system. No filer, quantizer and pooling synthesizer is used in this experiment.

• **simulated system**
The simulated system is a CMC generated at random $p(y_t|u_t, y_{t-1})$. Both input and output have ten levels, i.e., $m_{u^*} = m_{y^*} = 10$. The memory length is one.

• **setting of parameters**

1. forgetting factor for estimation $\lambda = 1$;

2. control horizon $H = 10$;

3. simulation length $T = 200$;
4. set point $y_t = 10$;
5. ideal controller is uniform distributed.

• **result**
The control aim of this experiment is to make the system output as close to 10 as possible.

The result is shown in Fig.9.4.

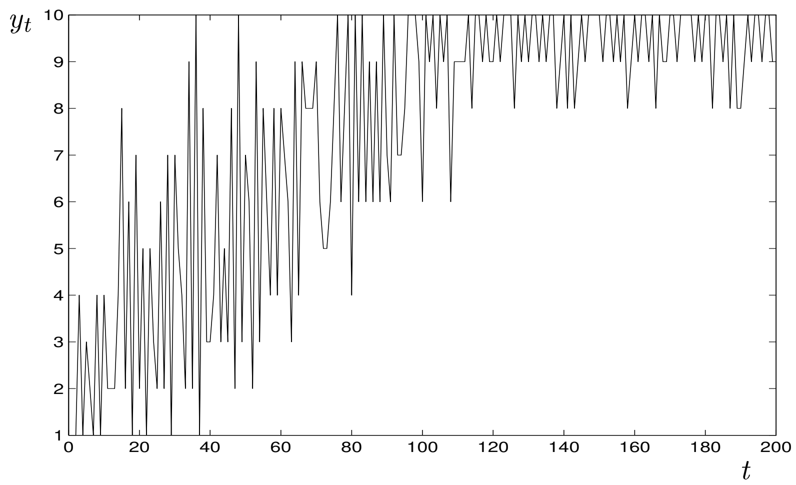

Figure 9.4: System Output

This figure says that system output starts to converge to the set point after 100 time instants. It means the probablistic control design works well.

9.4 Comprehensive Test

This experiment is designed to test the whole approximate control methodology on a coupled tanks system in order to see the performance of this methodolody on a real system.

• **coupled tanks system**
The description of the coupled tanks system can be found in Section 5.3.1. The dynamics of this system is modelled by an ARX model of the following structure

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + a_3 Y_{t-3} + b_0 U_t + b_1 U_{t-1} + b_2 U_{t-2} + c e_t$$

The coefficients of the model have the following values:

$$\begin{array}{lll} a_1 = 0.9717 & a_2 = 0.0072 & a_3 = 0.0231 \\ b_0 = 0.0042 & b_1 = 0.0009 & b_2 = 0.0010 \\ c = 0.0035 & & \end{array}$$

e_t is Gaussian white noise $N(0,1)$.

• **setting of parameters**

1. the number of filters $m = 3$.
2. the number of quantizers $n = 6$.
3. the forgetting factor for identification $\lambda = 0.99$.
4. the control horizon $H = 10$.
5. set point $Y_t = 0$.
6. the simulation length $T = 500$.

• **result**

The control aim of this experiment is to make the system output as close to zero as possible.

The result is shown in Fig.9.5. This figure says that the system output starts to

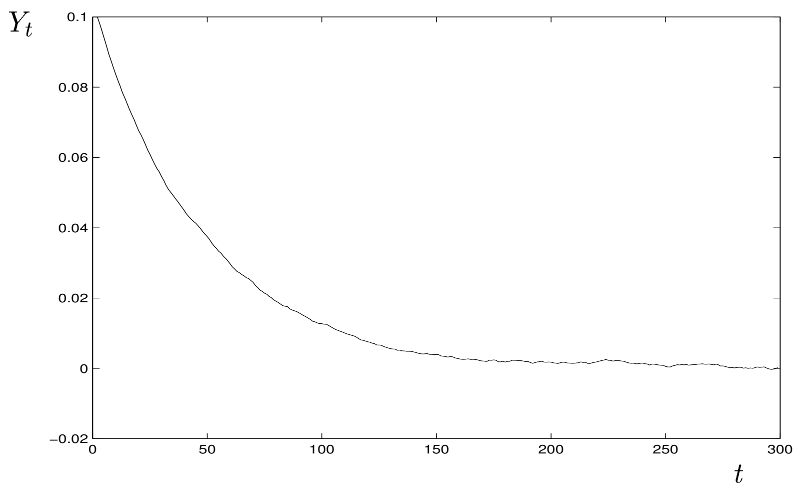

Figure 9.5: Output of Controlled Tank System

converge to the set point after 100 time instants. It means this approximate methodology achieves the expected goal and works properly. It indicates the feasibility of this approximate control methodology.

9.5 Summary

In this chapter, we study the performance of the approximate control methodology proposed in the previous chapter by four experiments. The first experiments illustrates the function of quantizers. The second experiment shows the efficiency of the filters. The third experiment tests the probabilistic control design. The last experiment tells us that this methodology is applicable to real systems. All the experiments in this chapter confirm the feasibility of the proposed approximate methodology.

Chapter 10

Conclusions

Estimation of linear ARX model and certainty-equivalence design are mostly used in contemporary adaptive controllers. In spite of the success of such controllers, they face difficulties whenever the controlled process is nonlinear or the range of inputs is restricted. Using a sort of gain schedulling a piece-wise linearization is often possible. The input restrictions, however, are coped with difficulties only. Thus, it is worth to search for an alternative class of adaptive controllers that take these restrictions into account.

In this thesis we try to complement ARX models by controlled Markov chains. By their nature, the gained controllers are able to cope both with nonlinear systems and restricted data ranges. However, dimensionality caused by the large state space and the long memory length is the main drawback which limits the applications of CMCs. This problem motivates the search for a feasible approximation.

An approximation methodology for Markov chain based prediction and adaptive control is proposed and developed in this thesis. It splits the original complex model into several simple ones. These simplified models are estimated independently and used for constructing approximate controllers. The approximate controllers are mixed into the overall controller by using Bayesian pooling theory. Moreover, plenty of experiments are presented in this thesis which further confirm the feasibility of this methodology. In this chapter the main contributions of this thesis are outlined and the open problems for further research are addressed.

10.1 Main Contributions

The main contribution of this thesis is to propose and develop an approximation methodology both for Markov chain-based prediction and adaptive control. Moreover, this methodology is tested by various experiments. The detailed description of the contributions is as follows.

1. An Approximate Prediction Methodology
   An approximate prediction methodology is proposed in Chapter 4. The main idea of this methodology is that we split the original model into a set of simple models with a small state space and a short memory length; then these simple models are estimated and corresponding predictors are obtained; the overall predictor is obtained by mixing these simple predictors together.

2. Validation of Prediction

   In Chapter 5, we investigate five experiments in order to test the performance of the proposed approximate prediction methodology on both simulated and real systems. Furthermore, the performance of these pooled predictors is compared with neural network. The results of these experiments demonstrate that this approximation methodology is applicable to nonlinear systems and MC-based predictors are comparable with NN-based ones. However, MC-based predictor is a better tool for on line mode.

3. Dual Control

   In Chapter 6, we apply the conventional dynamic programming method to control design of a controlled tossing of a coin in order to demonstrate complexity and basic features of dual control on a simple stochastic system.

   This result tells us that dynamic programming formally characterizes the ideal solution to adaptive control design of stochastic systems, but it fails to be applicable to realistic systems because of dimensionality drawback. An alternative control design is needed.

4. An Approximate Control Methodology

   In Chapter 8, the approximate prediction methodology is extended to the adaptive control case.

   The basic idea is that we apply several filters and quantizers to the original data so that a set of approximate models with small state spaces and short memory lengths are obtained; then these models are estimated and corresponding control design made; lastly, the inputs proposed by particular controllers are mixed into the final action applied.

5. Validation of Control

   In Chapter 9, various experiments are carried out in order to test the properties of filter, quantizer, probabilistic control design and the whole approximate control methodology. The results of these experiments indicate the feasibility of this approximation methodology.

10.2 Open Problems

The whole work in this thesis indicates that our approach to combat the curse of dimensionality is successful. In order to exploit this potential fully, we have to address the following open problems:

1. Improvement of the pooling theory such as the choice of the pre-prior distribution and the norm of the mixing vector.
2. Inclusion of prior information into prior statistics.
3. Reconsideration of dual control design.
4. Better filters and quantizers.
5. Testing (theoretical analysis, simulations, pilot plants, reality).
6. Extension to multivariate case.

Bibliography

[1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1947.

[2] A. Wald, *Statistical Decision Function*, John Wiley, New York, 1950.

[3] L.J. Savage, *The Foundations of Satistics*, John Wiley, New York, 1954.

[4] D. Balcwell and M.A. Girshick, *Theory of Games and Statistical Decision*, John Wiley, New York, 1954.

[5] H. Simon, *Theories of Decision-making in Economic and Behavioral Science*, Norton, New York, 1975.

[6] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[7] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, MIT Press, Cambridge, MA, 1949.

[8] F. Ling, D. Manolakis, and J.G. Proakis, "Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients", *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 837–845, 1986.

[9] J. Cioffi and T. Kailath, "Fast recursive-least-squares transversal filters for adaptive filtering", *IEEE Trans. on ASSP*, vol. 32, no. 2, pp. 304–337, 1984.

[10] M. Cioffi and T. Kailath, "Windowed fast transversal filters adaptive algorithms with normalization", *IEEE Trans. on ASSP*, vol. 33, no. 3, pp. 607–625, 1985.

[11] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters; a unified geometric theory", *IEEE Trans. on IT*, vol. 30, no. 2, pp. 222–236, 1984.

[12] B. Porat, B. Fridlander, and M. Morf, "Square-root covariance ladder algorithm", *IEEE Trans. on AC*, vol. 27, no. 4, pp. 813–829, 1982.

[13] G.N. Fouskitakis and S.D. Fassois, "On the estimation of long memory time series models", in *Electronic proceedings of ECC97*. Brussels, 1997.

[14] L. Berec and J. Rojíček, "Control Period Selection: Verification on Coupled Tanks", in *Preprints of European Control Conference ECC'97*, G. Bastin and M. Gevers, Eds. 1997.

[15] B. de Haas, V. Werbrouck, and V. Wertz, "Go control: a multivariable lqg controller for cement mills", in *Electronic proceedings of ECC97*. Brussels, 1997.

[16] V. Peterka, "Bayesian system identification", in *Trends and Progress in System Identification*, P. Eykhoff, Ed., pp. 239–304. Pergamon Press, Oxford, 1981.

[17] M. Kárný, "Towards fully probabilistic control design", *Automatica*, vol. 32, no. 12, pp. 1719–1722, 1996.

[18] P. J. Schweitzer, "A survey of aggregation-disaggregation in large Markov chains", in *Numerical Solution of Markov Chains*, W. J. Stewart, Ed., pp. 63–88. Marcel Dekker, New York, 1991.

[19] R. Kulhavý, *Recursive Nonlinear Estimation. A Geometric Approach*, Springer, London, 1996.

[20] R. Kulhavý, "System identification: from matching data to matching probabilities", in *Plenary Lectures and mini-courses of European Control Conference'97*, G. Bastin and M. Gevers, Eds., pp. 131–160. Brussels, 1997.

[21] R.A. Howard, *Dynamic Probabilistic System, Vol 1, Markov Models*, John Wiley, New York, 1971.

[22] J.G. Kemeny and J.L. Snell, *Finite Markov Chains*, Van Nostrand, Princeton, NJ, 1960.

[23] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol 1*, John Wiley, New York, 1957.

[24] M. Abramowitz and I.A. Stegun, *Handbook of mathematical functions*, Dover Publications, Inc., New York, 1972.

[25] R. Kulhavý, "Restricted exponential forgetting in real-time identification", *Automatica*, vol. 23, no. 5, pp. 589–600, 1987.

[26] S. Kullback, *Information Theory and Statistics*, Wiley, New York, 1959.

[27] R. Kulhavý, "A Kullback-Leibler distance approach to system identification", in *Preprints of the IFAC Symposium on Adaptive Systems in Control and Signal Processing*, Cs. Bányász, Ed., pp. 55–66. Budapest, 1995.

[28] M. Kárný, A. Halousková, and L. Zörnigová, "On pooling expert opinions", in *Preprints of the 10th IFAC Symposium on System Identification SYSID'94*, M. Blanke and T. Söderström, Eds., vol. 2, pp. 477–478,485–488. Danish Automation Society, Kopenhagen, 1994.

[29] L. Pavelková, "Approximate identification of Markov chains", in *Preprints of the European IEEE Workshop CMP'94*, L. Kulhavá, M. Kárný, and K. Warwick, Eds., pp. 335–340. ÚTIA AVČR, Prague, 1994.

[30] M. Kárný, A. Halousková, and P. Nedoma, "Recursive approximation by ARX model: a tool for Grey Box modelling", *International Journal of Adaptive Control and Signal Processing*, vol. 9, pp. 525–546, 1995.

[31] M. Kárný, A. Halousková, and L. Zörnigová, "Bayesian pooling of expert opinions", Tech. Rep. 1799, ÚTIA AV ČR, POBox 18, 182 08 Prague 8, Czech Republic, 1994, 21 pp.

[32] M. Kárný, M. Valečková, and H. Gao, "Mixtures of adaptive controllers based on Markov chains: a future of intelligent control?", in *Preprints of International Conference on Control 98*, vol. 1, pp. 721–726. The IEE, Swansea, 1998.

[33] L. Berec and M. Kárný, "Identification of reality in Bayesian context", in *Computer-Intensive Methods in Control and Signal Processing: Curse of Dimensionality*, K. Warwick and M. Kárný, Eds. Birkhauser, 1997.

[34] H. Gao and M. Kárný, "Can Markov chain supplement adaptive AR control", in *Preprints of the 2th European Workshop CMP'96*, L. Berec, J. Rojíček, M. Kárný, and K. Warwick, Eds., pp. 229–233. ÚTIA AVČR, Prague, 1996.

[35] M. Sláma, A. Procházka, and J. Šmíd, "Application of nonlinear time series modelling for meteorological data prediction", *Neural Network World*, vol. 5, no. 4, pp. 603–610, 1995.

[36] N.B. Karayiannis and A.N. Venetsanopoulos, *Artificial Neural Networks*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1993.

[37] A.A. Feldbaum, "Theory of dual control", *Automn Remote Control*, vol. 21, no. 9, pp. 874–880, 1033–1039, 1960.

[38] A.A. Feldbaum, "Theory of dual control", *Automn Remote Control*, vol. 22, no. 2, pp. 1–12, 109–121, 1960.

[39] K.J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[40] K.J Astrom, *Introduction to stochastic control*, Academic Press, New York, San Francisco, London, 1970.

[41] E. Nováková and Miroslav Kárný, "New approach to control design for Markov chains", in *European Control Conference. /CD-ROM/*, G. Bastin and M. Gevers, Eds., p. 221 kB. EUCA, Brussels, July 1997.