TECHNICAL UNIVERSITY OF CRETE
ELECTRONIC & COMPUTER ENGINEERING

Dissertation

ESTIMATION OF GENERAL IDENTIFIABLE STATE-SPACE MODELS

by

Christos A. Koniaris

Submitted in partial fullfilment of the
requirements for the degree of
Master of Science

September 2006

TECHNICAL UNIVERSITY OF CRETE

Department of

ELECTRONIC & COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Estimation of General Identifiable State-Space Models**" by **Christos A. Koniaris** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: September 2006

Supervisors:

Assoc. Prof. Alexandros Potamianos

Prof. Vassilios Digalakis

Reader:

Prof. Nicolaos Sidiropoulos

TECHNICAL UNIVERSITY OF CRETE

Date: **September 2006**

Author: **Christos A. Koniaris**

Title: **Estimation of General Identifiable State-Space Models**

Department: **Electronic & Computer Engineering**

Degree: **M.Sc.** Convocation: **September** Year: **2006**

Signature of Author

To Magda and my family.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Description
ANNs	Artificial Neural Networks
ASR	Automatic Speech Recognition
EM	Expectation Maximization
FAHHM	Factor Analyzed Hidden Markov Model
FFT	Fast Fourier Transform
GISSMs	Generalized Identifiable State Space Models
HAMM	Hidden Articulatory Markov Model
HDMs	Hidden Dynamic Models
HHMs	Hidden Markov Models
HTK	HMMs Toolkit
LDMs	Linear Dynamic Models
LDSM	Linear Dynamic Segment Model
LDSSMs	Linear Dynamic System Segment Models
ML	Maximum Likelihood
MFCCs	Mel Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
PDP	Parallel Distributed Processing
RTS	Rauch-Tung-Striebel
SMs	Segment Models

Acronym	**Description**
SNR	Signal to Noise Ration
SSM	Stochastic Segment Model
SLDS	Switching Linear Dynamical System
TDNN	Time Delay Neural Network
VQ	Vector Quantization

Abstract

The aim of this work is to investigate generalized forms of linear state-space dynamic models. Linear dynamic systems have been extensively used in the past for various applications as well as in speech recognition. In all cases, several modeling restrictions were applied to ensure that the model is identifiable. In this thesis, the generalization that has been proposed relaxes these constraints, and allows choosing full noise covariances and state vectors that have arbitrary increased dimension compared to the size of the observation vector. In addition, a canonical form of the system's matrices which ensures system's identifiability, has been used in order to perform this generalization. Furthermore, we investigate the use of an extra control input in the state equation. For all forms of linear dynamic models that we investigate we introduce novel maximum likelihood, element-wise, parameter estimation processes based on the Expectation-Maximization algorithm. In our experiments on artificial data, we show that we can obtain good estimates of the system's parameters for various system setups.

Furthermore, we applied the proposed system on a AURORA 2 digit classification task to evaluate its modeling capability. The majority of the *automatic speech recognition* (ASR) systems that have been extensively used, have been based on the *Hidden Markov Models* (HMMs). The use of HMMs seems to be successful and efficient, however they are based on a series of assumptions some of which are known to be poor. In particular, the assumption that each output is independent from the previous and only depends on the current state given the current state, gives little flexibility for modeling more complex dependencies. State-space models may be

used to address some shortcomings of this assumption. Linear state-space models are based on a continuous state vector evolving through time according to a state evolution process. The observations are then generated by an observation process (a linear dynamic model), which maps the current continuous state vector onto the observation space.

Finally, we have also applied our system on weather data performing a three-day weather forecasting, showing that our proposed system could be applied in various tasks, beyond speech recognition. In this case, the model has been used as a predictor, a task that linear dynamic models can also be successfully applied. Results shown that we can achieve good predictions, having only a few observations.

Acknowledgements

I would like to thank Alexandros Potamianos, my supervisor, for his many suggestions and constant support during my Masters studies. I express my gratitude for his guidance and unceasing motivation during the last three years. I am also thankful to Vassilios Digalakis for his valuable help having perfectly served as a co-supervisor at the last year of my studies. His experience and scientific background, has proven advantageous in my research.

Vassilios Diakoloukas shared with me his knowledge and experience in the area of speech recognition and provided many useful references and friendly encouragement.

I would like also to thank, the guys from Speech Processing and Dialog Systems Group, for their support and help. I shared the same office with most of them, and many pleasant and difficult times inside and outside the lab.

Of course, I am grateful to my parents for their patience and *love*. I hope that the joy of my graduation, would make them happy and proud.

Finally, I wish to thank Magda, for always being there when I needed her support and for cheering me up and encouraging me when I felt down during my extremely arduous studies. I would not have been able to achieve everything I did without her. I hope one day to be able to recompense for all the sacrifices she has been through, and make her life a lot more "easier".

Chania, Greece — Christos Koniaris
August 7, 2006

"... Για σένα έχω μιλήσει σε καιρούς παλιούς
με σοφές παραμάνες και μ΄ αντάρτες απόμαχους
από τι να ΄ναι που έχεις τη θλίψη του αγριμιού
την ανταύγεια στο μέτωπο του νερού του τρεμάμενου
και γιατί, λέει, να μέλει κοντά σου να ΄ρθω
που δεν θέλω αγάπη αλλά θέλω τον άνεμο
αλλά θέλω της ξέσκεπης όρθιας θάλλασας τον καλπασμό

και για σένα κανείς δεν είχε ακούσει
για σένα ούτε το δίκταμο ούτε το μανιτάρι
στα μέρη τ΄ αψηλά της Κρήτης τίποτα
για σένα μόνο δέχτηκε ο Θεός να μου οδηγεί το χέρι ..."

Οδυσσέας Ελύτης
"Το μονόγραμμα"

Chapter 1

Introduction

Right at the start we have to make a statement which you are certainly familiar with, but it is necessary to repeat it again and again. It is that science does not try to explain, nor searches for interpretations but primarily constructs models. A model is a mathematical construction, which supplemented with some verbal explanation, describes the observed phenomena. Such a mathematical construction is proved if and only if it works, that is it describes precisely a wide range of phenomena. Furthermore it has to satisfy certain aesthetic criteria, i.e., it has to be more or less simple compared to the described phenomena.

J. Von Neumann

In engineering, many problems based on mathematical models of the examined subjects. Constructing an appropriate model for each task of real processes is of great importance. At the same time, this is an extremely painful and demanding task to be involved with. Scientists and engineers that work in this area, are known as *Knowledge Engineers*. The choice of this name is not arbitrary; in order to construct

a model for a certain process one should have a deep knowledge on process' way of functioning and also to be able to manage and efficiently transfer this knowledge into a useful model.

In this thesis we are investigating new forms of linear dynamic models. A system is considered to be *linear* when it has the following form

$$x_i(k+1) = \sum_{j=1}^{n} a_{ij}(k)x_j(k) + w_i(k) \tag{1.0.1}$$

where $x_j(k)$ are the state variables of the system and the values $a_{ij}(k)$ are fixed parameters or coefficients of the system. The argument k, denotes time dependence, and this dependency is independent of the values assumed by the state variables [38]. On the other hand, the term *dynamic* refers to phenomena that are changing in time. The opposite of a dynamic phenomenon is static which refers to a steady state. Almost everything in nature is dynamic. In solid state physics for example, we know that crystals atoms have some dynamic aspects; this is why the behavior of the solids is changing in time depending on the external conditions i.e., temperature. Finally, a *model* of a system is a tool we apply to answer questions about the considering system without to do an experiment [37]. Models are also been used in every day life. For example, when a person is considered to be "friendly", one would have been able to answer the question of how this person would react when you ask him for an information. These types of models are the called *mental models* and are not formulated with mathematics. They are based on intuition and experience. In this work, where we are dealing with engineering subjects, we are using the *mathematical models*. In these, the relationships between quantities that can be observed in a system are described as mathematical relations. Mathematical formulations are also

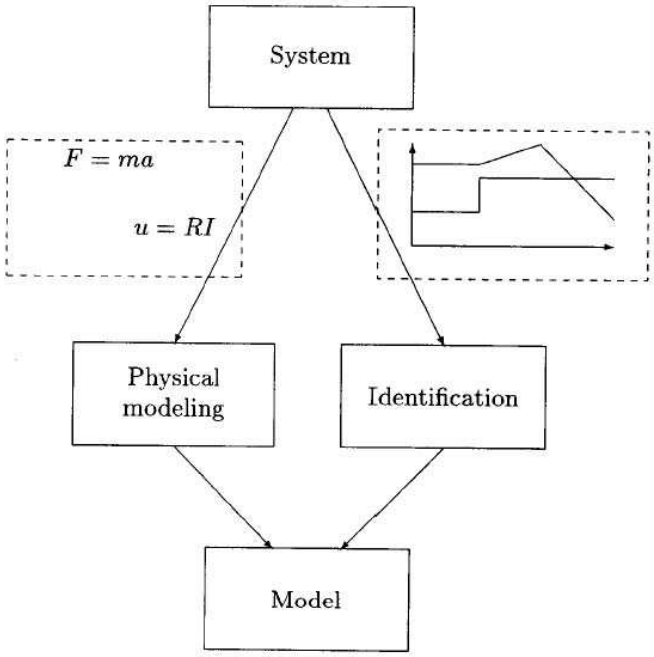

Figure 1.1: Model construction.

used in most of the natural laws in Physics. Figure (1.1), obtained by [37], show how a model is constructed.

Motivated by the research area of speech recognition, we initially started to investigate the construction of a model to be used for acoustic modeling. The proposed model is called Generalized Identifiable State Space Model (GISSM). As the name of our proposed model defines, it could be possible without to many changes, to be used in areas outside speech recognition. In this thesis we show how this could happen.

1.1 Thesis outline

The remainder of this dissertation is organized as follows. Chapter 2 gives a brief review of the Linear Dynamic Models. We deal with the use of LDMs in filtering and learning processes and present some estimation approaches written in literature.

In Chapter 3 we deal with the main work that has been done in this thesis. We begin by representing our model, the equations and the structure of it. We further discuss about our generalization and provide the proposed element-wise estimation method based on the EM algorithm. We examine two cases; a dynamic model without control input and the same model with an extra control. We show how the equations are being modified, and provide experimental results on artificial data comparing the performance of the two cases and to previous methods.

The next chapter 4, reviews some of the most popular approaches in acoustic modeling. As we are interesting in statistical approaches, we mainly focus on these and hence we discuss about *Hidden Markov Models, Artificial Neural Networks* and the general family of *Segment Models*. In addition, we show how the proposed model and our estimation method can be applied in a real speech recognition system. We give details on the modifications that we had to do in order the model to be applied in speech data and show experiments in a digit recognition task.

Chapter 5 showing how our system could be applied in a weather forecasting task. The nature of the proposed model is such, that we believe that it could be possible to be applied in tasks beyond speech recognition. In this chapter we train our model to predict the temperatures of the next few days.

Finally, in chapter 6, we discuss our conclusions and summarize the contributions of this dissertation to the fields of estimation theory, pattern recognition (*speech recognition*), and prediction (*weather forecasting*). In the end we discuss about future work and some possible extensions.

1.2 Declaration

A major part of the work presented in chapter 4, concerning the training and the classification of the LDSSM in our speech recognition application, was carried out by Georgios Tsontzos and Vassilios Diakoloukas. I would like to express my gratitude and appreciation to both of them.

Chapter 2

Estimation of Linear Dynamic Models

We must not forget that when radium was discovered no one knew that it would prove useful in hospitals. The work was one of pure science. And this is a proof that scientific work must not be considered from the point of view of the direct usefulness of it. It must be done for itself, for the beauty of science, and then there is always the chance that a scientific discovery may become like the radium a benefit for humanity.

M. Curie

In this chapter we present an introductory treatment of the family of Linear Dynamic Models.

2.1 Learning and Estimation

Dynamical models have many applications in a wide range of subjects. In some tasks, the values of hidden states are known and we only have to estimate them. In such cases, it could be possible for us to obtain prior knowledge of the system's parameters,

based on previous experience upon the system. In this case, the problem is known as *inference* or *filtering* or *smoothing*, and the task is to estimate the hidden state sequence based on the known model parameters and the given observations.

In some other applications, our knowledge and experience of the problem is poor, and we do not have any kind of information of the structure of the model as well as the observation and state evolution matrices. In this case, we give emphasis on learning a few parameters which model the observation data well. The problem is then called *learning* or *system identification* [62].

Filtering and smoothing have been extensively studied for continuous state models in signal processing [29, 30, 53, 54]. On the other hand, learning in continuous state models is the subject of a whole research area in Control theory called **System Identification**. Considering linear Gaussian models, there are several approaches to system identification [36]. In addition, the powerful EM algorithm for Linear dynamic systems derived originally in [66], and then reintroduced and extended in various applications, such as neural networks [22, 23] or speech recognition [13].

2.2 Parameter Estimation in LDMs

One of the most important statistical problems is the estimation of parameters in different dynamic models. Suppose that for a certain problem we have a few candidate models that could possible model the system in an efficient way. Suppose also that the above set of models has been parameterized using a parameter vector θ. The estimation of θ then, can be translated into the search of the optimum model within the set. There are some different approaches on how to get into the best model. In

this section we will deal with some of the most popular.

2.2.1 Maximum Likelihood estimation

A linear dynamic model (or Kalman Filter model or linear state-space model) is specified by the following pair of equations

$$x_{k+1} = Fx_k + w_k, \quad w_k \sim N(\epsilon, P) \tag{2.2.1}$$

$$y_k = Hx_k + v_k, \quad v_k \sim N(\lambda, R) \tag{2.2.2}$$

and an initial state distribution $x_1 \sim N(\mu_0, \Sigma_0)$. We use x_k and y_k to denote $n-$ and $m-$dimensional state and observation vectors respectively.

The Maximum Likelihood estimation [24] can be obtained by minimizing the following

$$J(\mathbf{Y}, \theta) = -L(\mathbf{Y}, \theta) = \sum_{k=0}^{N} \{\log |\Sigma_{e_k}(\theta)| + e_k^T(\theta)\Sigma_{e_k}^{-1}(\theta)e_k(\theta)\} + \text{constant} \tag{2.2.3}$$

where $L(\mathbf{Y}, \theta)$ denotes the log likelihood. In the above equation we used $e_k(\theta)$ and $\Sigma_{e_k}(\theta)$ to denote the prediction error and its covariance respectively. These quantities can be obtained by the Kalman filter equations [29]. In [31], the parametrization is being applied onto the innovations covariance Σ_e and Kalman gain K rather than P and R to simplify calculations.

2.2.2 Lower Bound on log-likelihood

When the state is hidden then the ML estimation may not be an efficient method. In this case the integral in the following equation, cannot be computed analytically.

$$L(\theta) = \log P(\mathbf{Y}|\theta) = \log \int_x P(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{X} \tag{2.2.4}$$

(In the above equation, P denotes distribution and should not be considered as the covariance of the noise in the state equation of the linear dynamic system.) However it could be computed using numerical methods. These methods are quite computationally expensive though. Hence, we use instead a *lower bound* on the log-likelihood [60].

Using any distribution Q over the hidden variables, we can obtain a lower bound [62] on log likelihood L

$$\begin{aligned}\log \textstyle\int_x P(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{X} &= \log \textstyle\int_x Q(\mathbf{X}) \frac{P(\mathbf{X},\mathbf{Y}|\theta)}{Q(\mathbf{X})} d\mathbf{X} \\ &\geq \textstyle\int_x Q(\mathbf{X}) \log \frac{P(\mathbf{X},\mathbf{Y}|\theta)}{Q(\mathbf{X})} d\mathbf{X} \\ = \textstyle\int_x Q(\mathbf{X}) \log P(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{X} &- \textstyle\int_x Q(\mathbf{X}) \log Q(\mathbf{X}) d\mathbf{X} \\ &= F(Q, \theta) \end{aligned} \tag{2.2.5}$$

where the inequality is known as Jensen's inequality. Using a Lagrange multiplier and an appropriate density function, we can then maximize the lower bound and the initial log-likelihood function [60]. The maximization of $F(Q, \theta)$ is equivalent to maximizing

$$G(Q(\mathbf{X})) = \lambda\left(1 - \int Q(\mathbf{X}) d\mathbf{X}\right) + F(Q(\mathbf{X}), \theta) \tag{2.2.6}$$

with respect to $Q(\mathbf{X})$.

2.2.3 Applying EM algorithm

If the data we have are all available and uncorrupted we could use maximum likelihood estimation. We can extend our application of maximum likelihood techniques to permit the learning of a linear dynamic system's parameters in situations where there

are missing or incomplete data [3, 10]. In our case, the state vector is considered to be the missing data. EM tries to maximize the following quantity at iteration p

$$Q(\theta^{(p+1)}|\theta^{(p)}) = E_{\theta}^{(p)}\Big\{L(X, Y, \theta^{(p+1)}|Y)\Big\} \tag{2.2.7}$$

where $L(X, Y, \theta)$ is given by

$$J(\mathbf{X}, \mathbf{Y}, \theta) = -L(\mathbf{X}, \mathbf{Y}, \theta) = \sum_{k=1}^{N}\Big\{\log|Q| + (x_k - Fx_{k-1})^T Q^{-1}(x_k - Fx_{k-1})\Big\} + \sum_{k=0}^{N}\Big\{\log|R| + (y_k - Hx_k)^T R^{-1}(y_k - Hx_k)\Big\} + \text{const} \tag{2.2.8}$$

In [66], an EM algorithm was presented for the case of linear dynamic systems in which the observation matrix H, is known. Many other researchers have also presented such parameter estimation methods based on EM [1, 32, 67]. In [13] however, a nontraditional method based on Expectation Maximization algorithm was presented, which could be considered as the continuous analog of the Baum-Welch estimation algorithm for hidden Markov models. The main idea was the consideration that the state vector of the linear system is observable, and that we want to find the ML estimates of the system parameters θ given Y and X.

The basis of the EM learning is to use the solutions to the filtering/smoothing problem to estimate the unknown hidden states given the observations and the current model parameters. Then use this complete data set to solve for new model parameters [62]. This process is repeated, using these new model parameters to infer the hidden states again and again until the convergence of the algorithm.

Chapter 3

General Identifiable State-Space Models

If human life were long enough to find the ultimate theory, everything would have been solved by previous generations. Nothing would be left to be discovered.

S. Hawking

Maximum Likelihood (ML) is the most well-known parameter estimation method that is widely used in statistical approaches. In order to perform ML estimation, we assume that the parameters of a pdf are fixed but unknown and we aim to find the set of parameters $\hat{\theta}$ that maximizes the likelihood of generating the observed data. A solution $\hat{\theta}$ could represent a true global maximum, a *local* maximum or minimum or (rarely) an inflection point of the log-likelihood function. If all solutions are found, we are guaranteed that one represents the true maximum, though we have to check each solution individually (or calculate second derivatives) to identify which is the global optimum [14].

This chapter describes the general identifiable state-space model. Initially, we

show the equations of the state and observation space and the structure of the model. *Linear Dynamic Models* (LDMs) have been extensively used in the past for various applications such as in speech recognition [12]. In all cases, several modeling restrictions were applied to ensure that the model is identifiable. Our generalization relaxes these constraints, and allows choosing full noise covariances and state vectors that have arbitrary increased dimension compared to the size of the observation vector. We incorporate the canonical form of the system's matrices proposed in [35] which ensures system's identifiability. Furthermore, we investigate the use of an extra control input in the state equation. Next, we introduce novel maximum likelihood, element-wise, parameter estimation processes based on the Expectation-Maximization algorithm. Finally, we show that we can obtain good estimates of the system's parameters for various system setups, using artificially generated data.

3.1 Introduction

The linear state-space dynamic system consists of the following pair of equations

$$x_{k+1} = Fx_k + w_k \tag{3.1.1}$$

$$y_k = Hx_k + v_k \tag{3.1.2}$$

where the state x_k at time k is a $(n \times 1)$ vector, the observation y_k is $(m \times 1)$ and w_k, v_k are uncorrelated, zero-mean Gaussian vectors with covariances

$$E\{w_k w_l^T\} = P\delta_{kl} \tag{3.1.3}$$

$$E\{v_k v_l^T\} = R\delta_{kl} \tag{3.1.4}$$

In the above equation δ_{kl} denotes the Kronecker delta and T denotes the transpose of a matrix. The initial state x_0 is Gaussian with known mean and covariance μ_0, Σ_0. Equation (3.1.1) describes the state dynamics, while (3.1.2) shows a prediction of the observation based on the state estimation.

3.2 Model Structure

The procedure of building mathematical models of dynamic systems based on observation measurements of the system is called *system identification*. The parametric structure of our multivariate state-space model has the following identifiable canonical form for the case in which x_k is a 9×1 vector and the observation vector y_k is a 3×1 vector.

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \times & \times & \times & \times & \times & \times & \times & \times & \times \end{bmatrix} \tag{3.2.1}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.2.2}$$

The number of rows with $\times$' s in F represents the free parameters of the matrix and equals the size of the output vector m. The ones in matrix H are highly correlated

to the number of the rows in F that are filled with free parameters, and also to the lines that these rows are placed in F.

To construct the form of the state transition matrix F we follow the process described in [35]. First we set its elements along the superdiagonal equal to one and the remaining elements are zeroed. Then, we choose arbitrarily the m row numbers r_i to be filled with free parameters where $i = 1, ..., m$. There is only the constraint that $r_m = n$ where m denotes the dimension of the observation and n the dimension of the state vector.

The observation matrix H is then constructed as follows. First, we define H to be $m \times n$ in size and filled with zeros. Assuming $r_0 = 0$, we let in each row $i = 1, ..., m$ of the matrix to have a one in column $r_{i-1}+1$. For instance, in the parametric structure shown in (3.2.1) and (3.2.2) we get:

$$r_1 = 3 \Rightarrow r_{1-1} + 1 = r_0 + 1 = 1$$

$$r_2 = 5 \Rightarrow r_{2-1} + 1 = r_1 + 1 = 4$$

$$r_3 = 9 \Rightarrow r_{3-1} + 1 = r_2 + 1 = 6.$$

When the control input is considered the linear system takes the following form:

$$x_{k+1} = Fx_k + Bu_k + w_k \tag{3.2.3}$$

$$y_k = Hx_k + v_k \tag{3.2.4}$$

The parametric structure of the control matrix B is then filled with free parameters:

$$\mathbf{B} = \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix} \tag{3.2.5}$$

The input control vector u_k is a $(l \times 1)$ deterministic vector, defined a-priori randomly.

3.3 Element-wise estimation with EM algorithm

3.3.1 Estimation without control input

For the system without control input , we try to find the ML estimates of its parameters θ given $\mathbf{Y} = [y_0\, y_1 \ldots y_N]$ and $\mathbf{X} = [x_0\, x_1 \ldots x_N]$. The ML estimates of θ are obtained by minimizing the following quantity

$$J(\mathbf{X}, \mathbf{Y}, \theta) = -L(\mathbf{X}, \mathbf{Y}, \theta) = \sum_{k=1}^{N} \Big\{ \log|P| + (x_k - Fx_{k-1})^T P^{-1} (x_k - Fx_{k-1}) \Big\}$$
$$+ \sum_{k=0}^{N} \Big\{ \log|R| + (y_k - Hx_k)^T R^{-1} (y_k - Hx_k) \Big\} + \text{const} \tag{3.3.1}$$

since, without loss of generality, w_k and v_k were assumed uncorrelated white Gaussian noise sources. In the appendix A.1 is shown that the estimates of the system's

parameters are given by

$$\hat{F}^{i,j} = \frac{\sum_{c=1}^{M}\left\{(cof(\hat{P}^{i,c}))(\Gamma_4^{c,j})\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1,c\neq i}^{M}\left\{(cof(\hat{P}^{i,c}))(\hat{F}^{c,j})(\Gamma_3^{j,j})\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\left\{(cof(\hat{P}^{i,c}))\sum_{r=1,r\neq j}^{M}\left\{(\hat{F}^{c,r})(\Gamma_3^{r,j})\right\}\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} \tag{3.3.2}$$

$$\hat{P}^{i,j} = (\Gamma_2^{i,j}) - \sum_{r=1}^{M}(\hat{F}^{i,r})(\Gamma_4^{j,r}) - \sum_{r=1}^{M}(\hat{F}^{j,r})(\Gamma_4^{i,r}) + \sum_{c=1}^{M}\sum_{r=1}^{M}(\hat{F}^{i,c})(\hat{F}^{j,r})(\Gamma_3^{c,r}) \tag{3.3.3}$$

$$\hat{R} = \Gamma_5 - \Gamma_6\Gamma_1^{-1}\Gamma_6^T \tag{3.3.4}$$

where $cof(\hat{P}^{i,c})$ is the cofactor of the element $\hat{P}^{i,c}$ of the covariance P. Index i denotes the $i-th$ row of a matrix, and j denotes the $j-th$ column, and M denotes the column dimension of F. The sufficient statistics [12, 13] that involved in the previous equations are given by

$$\Gamma_1 = \frac{1}{N+1}\sum_{k=0}^{N} x_k x_k^T \tag{3.3.5}$$

$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N} x_k x_k^T \tag{3.3.6}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N} x_{k-1} x_{k-1}^T \tag{3.3.7}$$

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N} x_k x_{k-1}^T \tag{3.3.8}$$

$$\Gamma_5 = \frac{1}{N+1}\sum_{k=0}^{N} y_k y_k^T \tag{3.3.9}$$

$$\Gamma_6 = \frac{1}{N+1}\sum_{k=0}^{N} y_k x_k^T. \tag{3.3.10}$$

The statistics shown above require the following quantities at iteration p:

$$E_{\theta^{(p)}}\{y_k x_k^T|\mathbf{Y}\} = y_k \hat{x}_{k|N} \tag{3.3.11}$$

$$E_{\theta^{(p)}}\{y_k y_k^T|\mathbf{Y}\} = y_k y_k^T \tag{3.3.12}$$

$$E_{\theta^{(p)}}\{x_k x_{k-1}^T|\mathbf{Y}\} = \Sigma_{k,k-1|N} + \hat{x}_{k|N}\hat{x}_{k-1|N}^T \tag{3.3.13}$$

$$E_{\theta^{(p)}}\{x_k x_k^T|\mathbf{Y}\} = \Sigma_{k|N} + \hat{x}_{k|N}\hat{x}_{k|N}^T. \tag{3.3.14}$$

Equations (3.3.2) through (3.3.4) form the Maximization step of the EM algorithm. For the Expectation step of the EM algorithm we need to compute the required statistics, and we use the fixed interval smoothing form of the Kalman filter (RTS smoother) [54]. It consists of a backward pass that follows the standard Kalman filter forward recursions [29]. In addition, we computed also the cross-covariances proposed by Digalakis [12] in both the forward and the backward pass. All the necessary recursions are shown below.

Forward Recursions

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k \tag{3.3.15}$$

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k} \tag{3.3.16}$$

$$e_k = y_k - H\hat{x}_{k|k-1} \tag{3.3.17}$$

$$K_k = \Sigma_{k|k-1}H^T\Sigma_{e_k}^{-1} \tag{3.3.18}$$

$$\Sigma_{e_k} = H\Sigma_{k|k-1}H^T + R \tag{3.3.19}$$

$$\Sigma_{k|k} = \Sigma_{k|k-1} - K_k\Sigma_{e_k}K_k^T \tag{3.3.20}$$

$$\Sigma_{k,k-1|k} = (I - K_kH)F\Sigma_{k-1|k-1} \tag{3.3.21}$$

$$\Sigma_{k+1|k} = F\Sigma_{k|k}F^T + P \tag{3.3.22}$$

Backward Recursions

$$\hat{x}_{k-1|N} = \hat{x}_{k-1|k-1} + A_k[\hat{x}_{k|N} - \hat{x}_{k|k-1}] \tag{3.3.23}$$

$$\Sigma_{k-1|N} = \Sigma_{k-1|k-1} + A_k[\Sigma_{k|N} - \Sigma_{k|k-1}]A_k^T \tag{3.3.24}$$

$$A_k = \Sigma_{k-1|k-1}F^T\Sigma_{k|k-1}^{-1} \tag{3.3.25}$$

$$\Sigma_{k,k-1|N} = \Sigma_{k,k-1|k} + [\Sigma_{k|N} - \Sigma_{k|k}]\Sigma_{k|k}^{-1}\Sigma_{k,k-1|k} \tag{3.3.26}$$

To summarize, at each iteration we compute the sufficient statistics described previously using the above recursions of the Forward-Backward and the old estimates of the model parameters (E-step). Then, the new estimates can be obtained from these statistics via the proposed element-wise way (M-step).

3.3.2 Estimation with additional control input

Assuming that the additional control term Bu_k is inserted in the state equation, which becomes as in (3.2.3), the ML estimates of θ are then obtained by minimizing:

$$\begin{aligned} J(\mathbf{X},\mathbf{Y},\theta) = -L(\mathbf{X},\mathbf{Y},\theta) = \sum_{k=1}^{N}\Big\{\log|P| \\ +(x_k - Fx_{k-1} - Bu_{k-1})^T P^{-1}(x_k - Fx_{k-1} - Bu_{k-1})\Big\} \\ +\sum_{k=0}^{N}\Big\{\log|R| + (y_k - Hx_k)^T R^{-1}(y_k - Hx_k)\Big\} + \text{constant} \end{aligned} \tag{3.3.27}$$

The element-wise estimates of the parameters are now given by:

$$\begin{aligned} \hat{F}^{i,j} = \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))(\Gamma_4^{c,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{q=1}^{T}\Big\{(\hat{B}^{c,q})(\Gamma_8^{q,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \\ \frac{\sum_{c=1,c\neq i}^{M}\Big\{(cof(\hat{P}^{i,c}))(\hat{F}^{c,j})(\Gamma_3^{j,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{r=1,r\neq j}^{M}\Big\{(\hat{F}^{c,r})(\Gamma_3^{r,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} \end{aligned} \tag{3.3.28}$$

$$\begin{aligned} \hat{B}^{i,j} = \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))(\Gamma_{10}^{c,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{s=1}^{M}\Big\{(\hat{F}^{c,s})(\Gamma_7^{s,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \\ \frac{\sum_{c=1,c\neq i}^{M}\Big\{(cof(\hat{P}^{i,c}))(\hat{B}^{c,j})(\Gamma_9^{j,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{q=1,q\neq j}^{T}\Big\{(\hat{B}^{c,q})(\Gamma_9^{q,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} \end{aligned} \tag{3.3.29}$$

$$\hat{P}^{i,j} = (\Gamma_2^{i,j}) - \sum_{r=1}^{M}(\hat{F}^{i,r})(\Gamma_4^{j,r}) - \sum_{r=1}^{M}(\hat{F}^{j,r})(\Gamma_4^{i,r}) + \sum_{c=1}^{M}\sum_{r=1}^{M}(\hat{F}^{i,c})(\hat{F}^{j,r})(\Gamma_3^{c,r}) - \sum_{q=1}^{T}(\hat{B}^{i,q})(\Gamma_{10}^{j,q}) + \sum_{q=1}^{T}\sum_{r=1}^{M}(\hat{B}^{i,q})(\hat{F}^{j,r})(\Gamma_8^{q,r}) - \sum_{q=1}^{T}(\hat{B}^{j,q})(\Gamma_{10}^{i,q}) + \sum_{r=1}^{M}\sum_{q=1}^{T}(\hat{F}^{i,r})(\hat{B}^{j,q})(\Gamma_7^{r,q}) + \sum_{q=1}^{T}\sum_{p=1}^{T}(\hat{B}^{i,q})(\hat{B}^{j,p})(\Gamma_9^{q,p}) \quad (3.3.30)$$

$$\hat{R} = \Gamma_5 - \Gamma_6\Gamma_1^{-1}\Gamma_6^T \quad (3.3.31)$$

where T denotes the column size of B.

The additional sufficient statistics that we need for the case of control input, are given by

$$\Gamma_7 = \frac{1}{N}\sum_{k=1}^{N} x_{k-1}u_{k-1}^T \quad (3.3.32)$$

$$\Gamma_8 = \frac{1}{N}\sum_{k=1}^{N} u_{k-1}x_{k-1}^T \quad (3.3.33)$$

$$\Gamma_9 = \frac{1}{N}\sum_{k=1}^{N} u_{k-1}u_{k-1}^T \quad (3.3.34)$$

$$\Gamma_{10} = \frac{1}{N}\sum_{k=1}^{N} x_k u_{k-1}^T \quad (3.3.35)$$

The statistics shown above require the following quantities at iteration p:

$$E_{\theta^{(p)}}\{x_{k-1}u_{k-1}^T|\mathbf{Y}\} = \hat{x}_{k-1|N}u_{k-1}^T \quad (3.3.36)$$

$$E_{\theta^{(p)}}\{u_{k-1}x_{k-1}^T|\mathbf{Y}\} = u_{k-1}\hat{x}_{k-1|N}^T \quad (3.3.37)$$

$$E_{\theta^{(p)}}\{u_{k-1}u_{k-1}^T|\mathbf{Y}\} = u_{k-1}u_{k-1}^T \quad (3.3.38)$$

$$E_{\theta^{(p)}}\{x_k u_{k-1}^T|\mathbf{Y}\} = \hat{x}_{k|N}u_{k-1}^T. \quad (3.3.39)$$

The Forward-Backward recursions are the same as shown in the previous paragraph. The only difference is at the computation of $\hat{x}_{k+1|k}$, which its now given by $\hat{x}_{k+1|k} = J\hat{z}_{k|k}$, where $J = [F\ B]$.

3.4 Experiments with artificial data

We applied the proposed estimation method using as training data a Gaussian random sequence generated artificially by a predefined system. The parameters of the predefined system were chosen randomly, based on the canonical forms shown in (3.2.1), (3.2.2) and (3.2.5), as well as square symmetric structures for the noise covariances. We then ensure system's stability, observability and controllability. When the state equation includes a control input we also set the control vector u_k randomly from a zero mean Gaussian distribution. The control input and the training data are considered a known deterministic quantity in the training process. Based on these data and the EM algorithm, we expect to estimate system parameters close to the predefined ones.

For the initialization of the EM algorithm we choose randomly an initial set of system's parameters. We only check to ensure that the system based on this initial configuration is observable, controllable and stable. Initialization is very important since it can influence the convergence of the algorithm. In our experiments, various initial sets of parameters highly affected the convergence rate of the estimation process.

Using the initial conditions and the training data we perform 50 iterations of the expectation and maximization steps of the EM algorithm. In the expectation step we

compute the forward - backward counts from the Kalman smoother and collect the sufficient statistics. In the maximization step we obtain new estimates for the system parameters based on the element-wise process that we propose. Since the estimates of the parameters of the state linear equation F, B and P are mutually dependent, we might consider an additional iterative estimation process of these parameters based on the same sufficient statistics obtained in the expectation step. This iterative process can be terminated when a predefined threshold of the distance between two successive estimates is reached. The distance metric that we consider between any set of parameter estimations θ and $\hat{\theta}$ is norm 1 described as

$$D(\theta, \hat{\theta}) = \frac{\|\theta - \hat{\theta}\|}{\|\theta\|}. \tag{3.4.1}$$

In the first set of experiments we consider 1000 artificially generated training data and the same dimension for the state and observation vector ($n = m = 3$). We can therefore compare the performance of our element-wise ML estimation process in figure (3.2) to the full matrix estimation process proposed in [12, 13] in figure (3.1). In both experiments the same predefined system and the same EM initialization is considered. The results include the increase of the log-likelihood versus the number of EM iterations. The increase of the log-likelihood does not guarantee convergence to a global maximum since local optima might be reached. However, if a global maximum is estimated in a Maximum Likelihood fashion, it is guaranteed to be the optimum estimate [10].

It is therefore necessary to show the distances between the estimated parameters $\hat{\theta}_{ML} = (\hat{F}_{ML}, \hat{P}_{ML}, \hat{R}_{ML})$ and the actual ones of the predefined system for various number of EM iterations to check actual convergence to the true values. The distance is calculated based on the distance metric defined in (3.4.1). It can be seen that the

estimation with the proposed element-wise method shows similar convergence rates to the real values compared to the full estimation. Similar findings can be obtained from tables (3.1), (3.2), and (3.3) where we present the estimated values for F, P, and R respectively after 50 iterations with both element-wise and full estimation methods.

	Estimation for F
Actual	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.8 & -1.79 & 1.9 \end{pmatrix}$
Full estimation	$\begin{pmatrix} -0.0001 & 1.0048 & -0.0254 \\ 0.0386 & -0.0443 & 1.0291 \\ 0.8173 & -1.8161 & 1.9389 \end{pmatrix}$
Element-wise	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.7881 & -1.7815 & 1.9044 \end{pmatrix}$

Table 3.1: Estimation of F with statistics and element-wise method for $n = m = 3$

A second set of experiments which compares element-wise estimation of the system parameters, with control input in figure (3.4), and without control in figure (3.3). We consider increased state size ($n = 5$) compared to the size of the observation vector ($m = 2$) and 1000 training sentences. It can be seen that the additional control term results in faster convergence to the real parameter values. An issue worth-mentioning, is that the distance does not always decreases as the number of iterations increases. This is attributed to the mutual dependence of the system's parameters. The ML estimation of a parameter might temporarily converge to a local optimum. However, as the remaining parameters converge to their true values they can influence the

	Estimation for P
Actual	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
Full estimation	$\begin{pmatrix} 1.086 & 0.1387 & 0.0429 \\ 0.1387 & 1.0403 & -0.1126 \\ 0.0429 & -0.1126 & 0.8447 \end{pmatrix}$
Element-wise	$\begin{pmatrix} 1.1334 & 0.1382 & 0.0292 \\ 0.1382 & 1.0206 & -0.0775 \\ 0.0292 & -0.0775 & 1.0187 \end{pmatrix}$

Table 3.2: Estimation of P with statistics and element-wise method for $n = m = 3$

convergence of this parameter to another local or global optimum. If we compare figures (3.4) and (3.3) we can conclude that adding the control input the convergence towards the global optimum is smoother. Similar findings are shown in figures (3.5) and (3.6) where the state and observation size was set to $n = 9$ and $m = 3$ respectively. Once again the extra input helped the system to reach more close to the real values.

	Estimation for R
Actual	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
Full estimation	$\begin{pmatrix} 0.9503 & 0.1093 & 0.1260 \\ 0.1093 & 1.0181 & -0.1400 \\ 0.1260 & -0.1400 & 0.8802 \end{pmatrix}$
Element-wise	$\begin{pmatrix} 0.9023 & 0.0787 & 0.0901 \\ 0.0787 & 1.0312 & -0.1193 \\ 0.0901 & -0.1193 & 0.9326 \end{pmatrix}$

Table 3.3: Estimation of R with statistics and element-wise method for $n = m = 3$

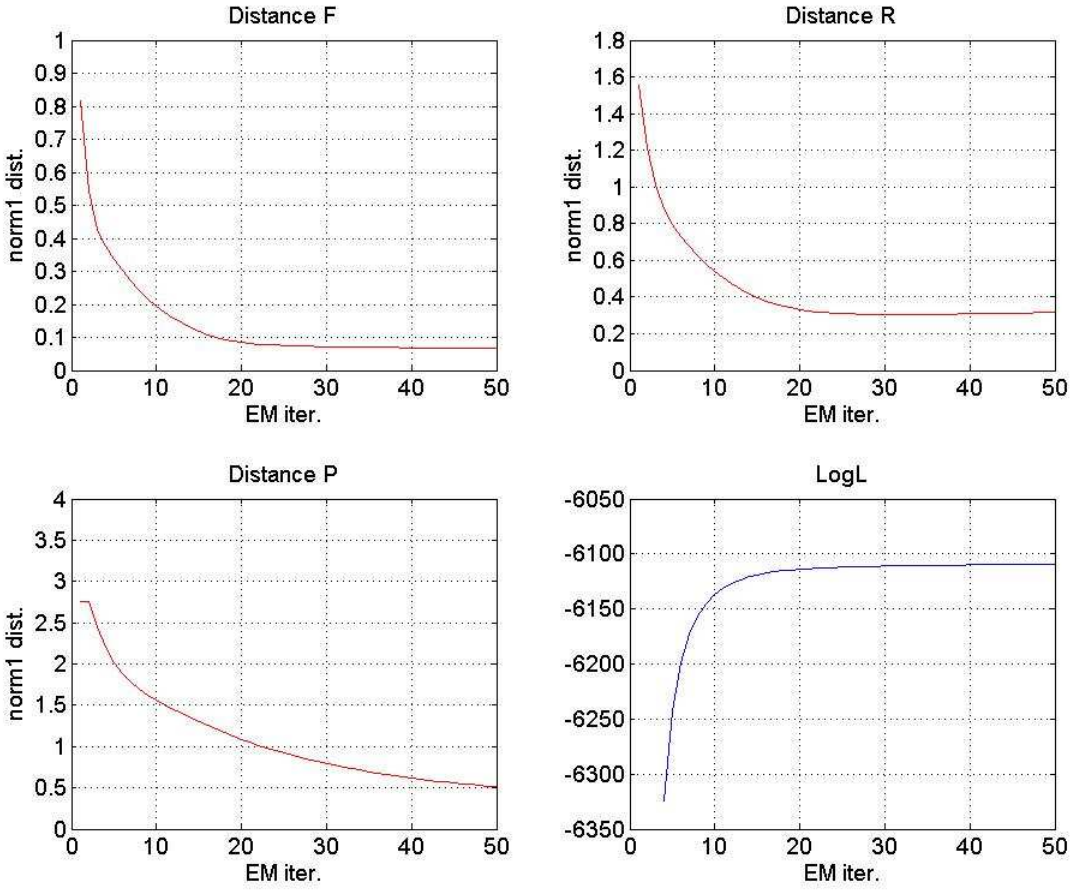

Figure 3.1: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=3 and m=3. (full matrix estimation)

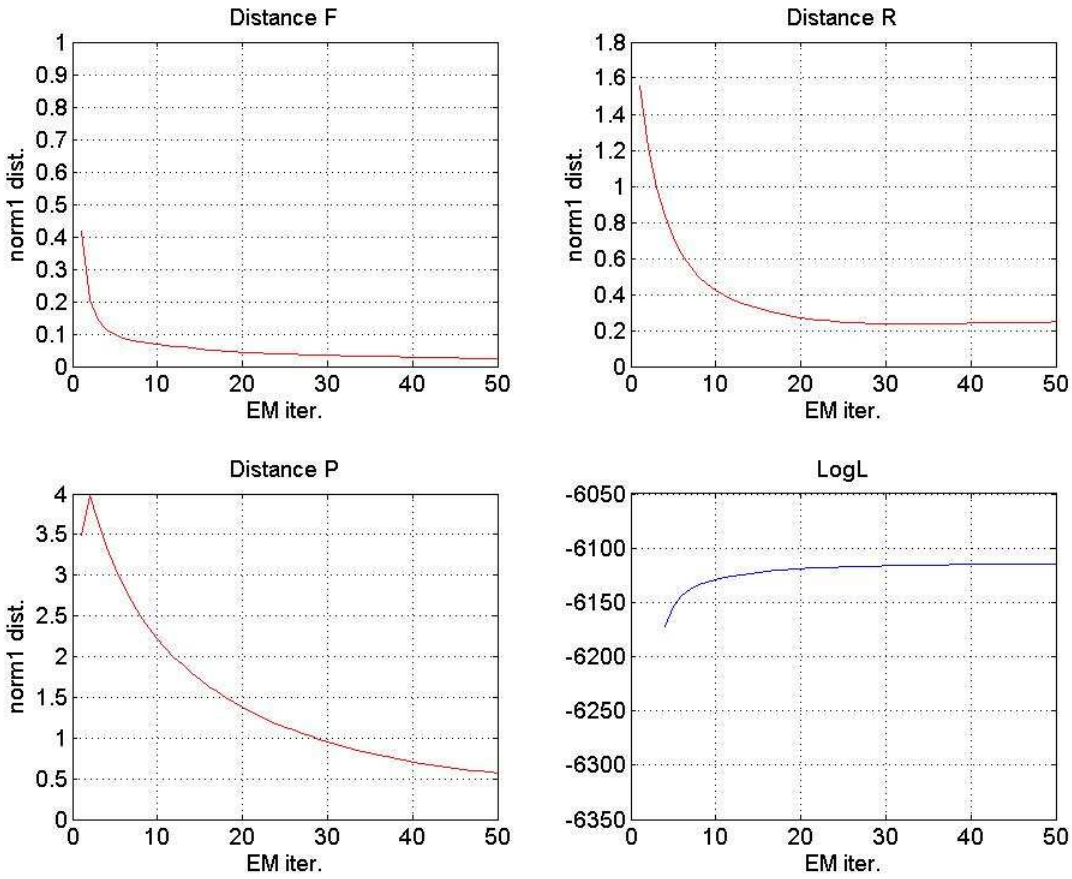

Figure 3.2: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=3 and m=3. (element-wise estimation)

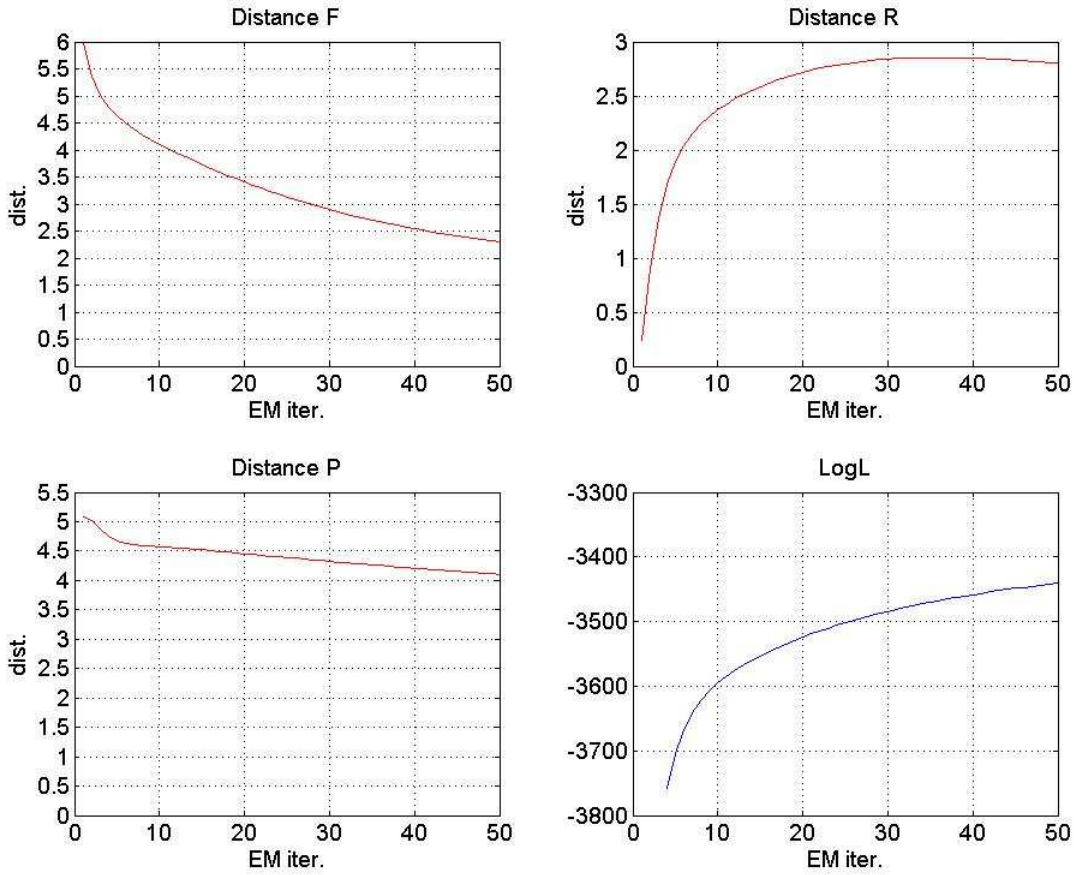

Figure 3.3: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=5 and m=2. (element-wise estimation)

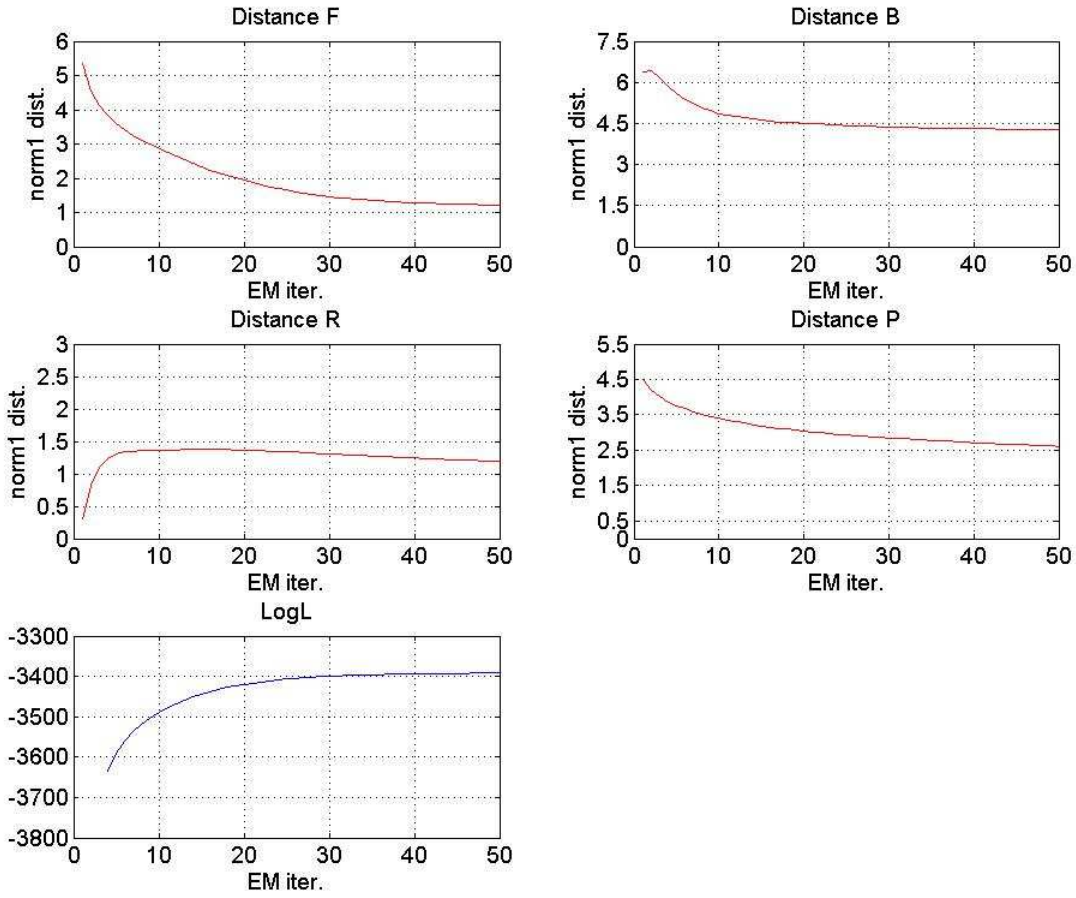

Figure 3.4: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=5 and m=2. (element-wise estimation with additional control input)

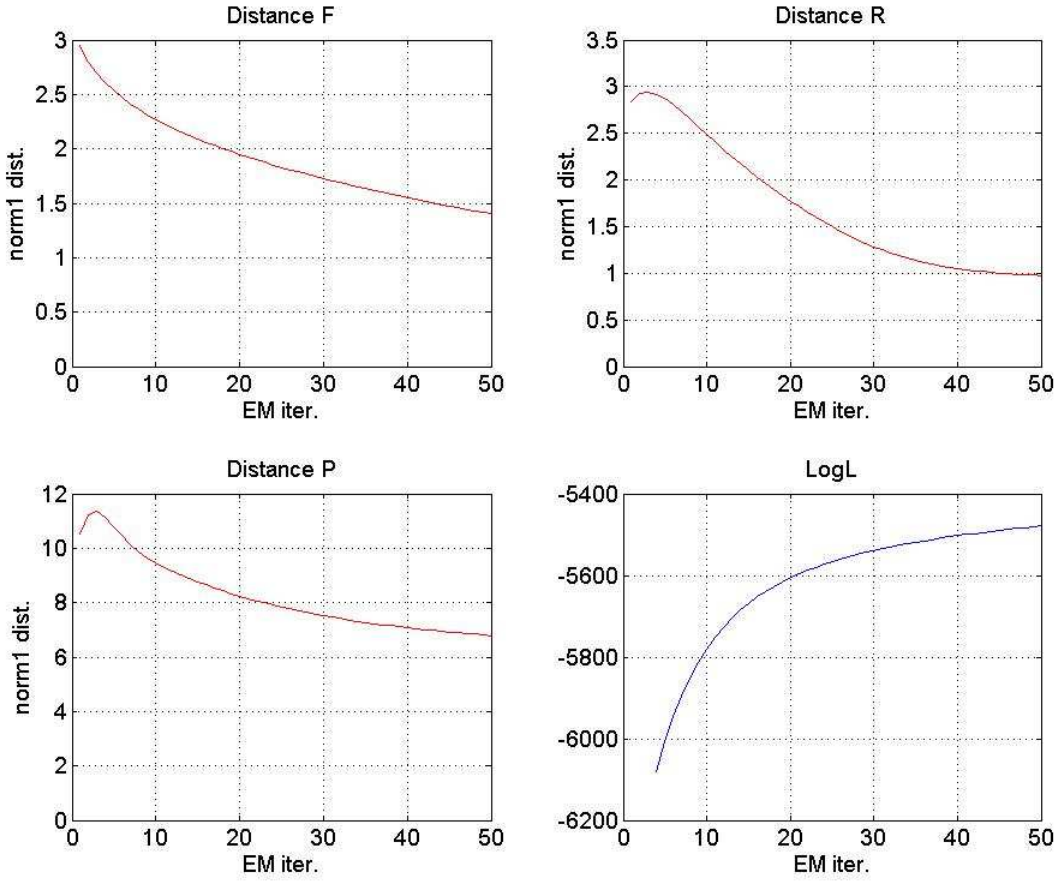

Figure 3.5: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=9 and m=3. (element-wise estimation)

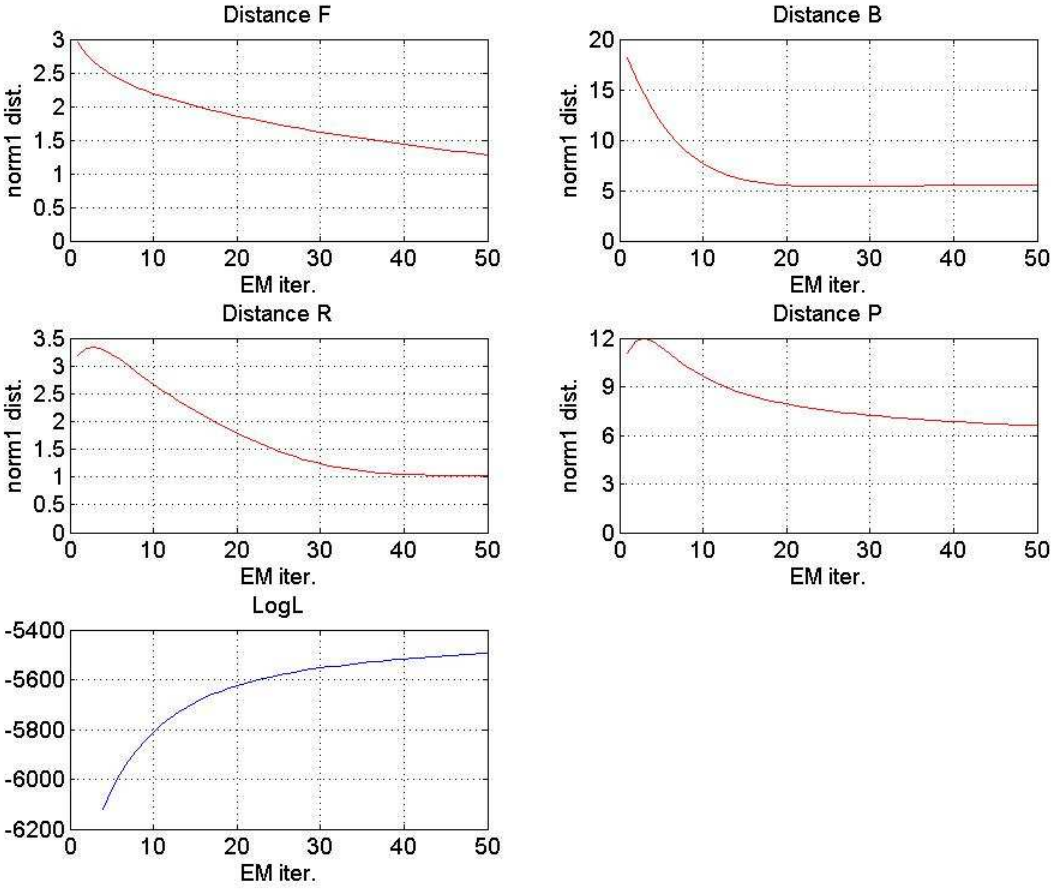

Figure 3.6: Distances between the estimated matrices F, P, R and their actual values. Log-likelihood is also shown. Case n=9 and m=3. (element-wise estimation with additional control input)

3.5 Discussion

We presented a novel maximum-likelihood algorithm for the re-estimation in an element-wise manner of the parameters of a generalized linear dynamic system. Most of the current systems follow a constrained structure which considers diagonal noise covariances, full state transition matrices under the assumption of identical state and observation dimensionality and the lack of control input. In the generalized system that we investigated, although we relax all these constraints, we still retain its identifiability. Experiments in artificial data, show that our algorithm is able to efficiently converge to the real values. The addition of the control input, enhances system performance and improves convergence speed. It also seems to overcome problems such as entrapment in a local optimum. Our generalized scheme can prove to be beneficial in various applications since it increases the degrees of freedom.

Chapter 4

Application on Speech: digit recognition

> *It is impossible to know all the causal factors. We can only choose a limited number of significant factors, and use these for predicting future events, being forced to ignore factors having only minor influence.*
>
> *H. Reichenbach*

Since the very early years of humanity, speech communication has been, and will continue to be, the dominant mode of human social bonding and information exchange. This behavior has a natural reflection on the way humans prefer to interact with technological artifacts, such as computers. Speech scientists and engineers has been working in the area of *speech communication* between humans and computers for over six decades. A major task in speech science is *speech recognition*. In this, the goal is to develop a device that transcribes human speech into written text at the same level of accuracy as, or higher than, that exhibited by humans.

A few years ago, communicate with a computer or a machine in general via speech was a part of science-fiction filmography. In nowadays, *Automatic Speech Recognition*

(ASR) has moved from that stage to daily reality, at least for people who live in developed countries. Applications of this technology can be found in mobile phones, in cars, in security buildings, in hospitals or in online airplane tickets booking. People with specialities have also been profited by speech recognition devices.

The most popular approaches to the speech recognition problem today are based on statistical methods. In this chapter we shall present different approaches for acoustic modeling. At the beginning we shall give the general definition of the acoustic model and then some of the most popular forms of it that are widely used by scientists. Next, we shall present how the proposed system could be applied as acoustic model in a speech recognition task. The data that have been used are digits spoken in a natural way by native speakers.

4.1 Acoustic Modeling Overview

An automatic speech recognition system mainly consists of the blocks shown in figure (4.1), obtained by [60]. These are the *front-end*, the *acoustic models*, the *language model*, the *lexicon* and the *search algorithm* [60]. Acoustic Modeling plays a critical role in a speech recognition system and is very important in improving accuracy. Consider a sequence of speech vectors or *observations* $\boldsymbol{O}$, defined as $O = o_1, o_2, ..., o_T$ where o_t is the speech vector observed at time t. The goal of speech recognition is to find out the corresponding word sequence $W = w_1, w_2, ..., w_T$ that has the maximum a-posteriori probability $P(W|O)$

$$\hat{W} = \arg\max P(W|O) = \frac{P(O|W)P(W)}{P(O)} \tag{4.1.1}$$

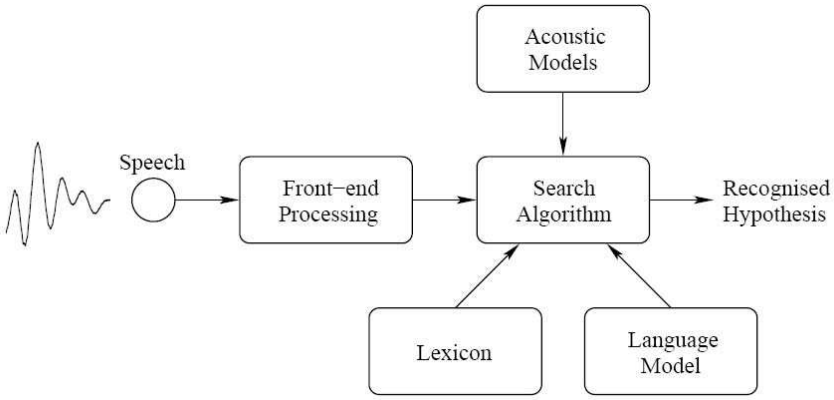

Figure 4.1: A real speech recognition system

The above formula is known as *Bayes' rule*. It should be noted that the likelihood of the observation sequence, $P(O)$, may be omitted in the maximization since it is independent of the word sequence. The conditional likelihood $P(O|W)$ is called the *acoustic model* and the $P(O)$ is called the *language model.*

In practice, the problem we need to overcome is to build robust acoustic models in order to decode (recognize) the spoken utterance. For small-vocabulary applications, the unit that usually being modeled is word, however for large-vocabulary speech recognition tasks, we need to decompose the word into some sub-word units, which called *phonemes.* In all cases, an optimal acoustic model has to reflect the speech production mechanism as naturally as it can, and to be able to model contextual effects such as co-articulation.

Hidden Markov Models (HMMs) and *Artificial Neural Networks* (ANNs) are the most popular approaches to acoustic modeling. Both are stochastic methods and especially the first one has become a "standard" in speech recognition. In the last 15 years though, *Segment-based Models* (SMs) have been developed from a variety

of research groups all over the world. These models seem to overcome some of the problems we meet in HMMs and ANNs. In the following, we shall represent each one of these different approaches for the acoustic model.

4.2 Hidden Markov Models

Hidden Markov Models are the most flexible and successful statistical approach and also very popular for acoustic modeling in speech recognition [2, 28, 52]. In HMMs, it is assumed that the sequence of observed speech vectors corresponding to each word or phone is generated by a Markov model [69] as shown in figure (4.2). Hence, this

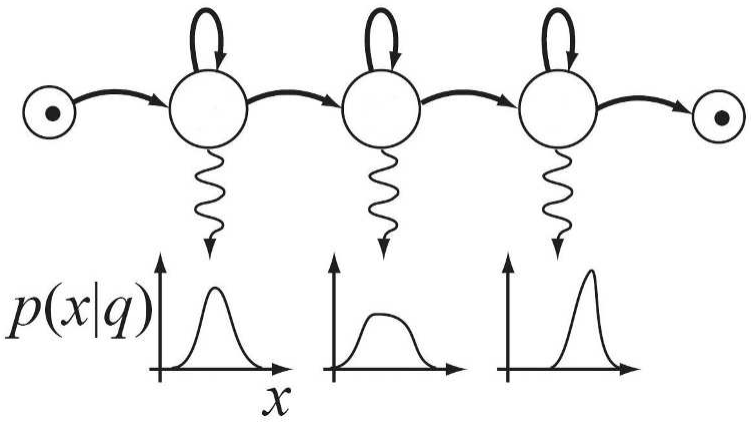

Figure 4.2: A hidden Markov model

model can be viewed as a double-embedded stochastic process with an underlying stochastic process (the state sequence) not directly observable - the name "hidden" has been adopted due to this fact. This hidden process can only be probabilistically associated with another, observable stochastic process which produces the sequence of features we observe [27].

Formally an HMM consist of the following elements:

• Number of states: N

• Number of distinct observation symbols: M for discrete HMMs and ∞ for continuous HMMs

• State transition probability distribution: $\alpha_{i,j}$

• Output distribution of state j: O_j

• Initial state probability: π_i.

To summarize, a complete specification of an HMM includes two constant parameters, N and M, that represents the total number of states and the size of observation alphabets respectively, and three sets (matrices) of probability measures A, O, and π. For convenience, we use the following notation

$$\lambda = (A, O, \pi) \tag{4.2.1}$$

to denote the whole parameter set of an HMM [27].

4.2.1 Types of HMMs

HMMs can be suitable classified according to the nature of the elements of the O matrix, which are distribution functions [19]. If observations are vectors of symbols in a finite alphabet of N different elements then, the distributions are defined on finite spaces, and the HMMs are called *discrete HMMs.* If the observation does not come from a finite set, but from a continuous space, the discrete output distribution discussed above needs to be modified. In this case, we have to imposed a series of limitations on the functional form of the distributions, in order to have a logical number of statistical parameters to estimate. The approach that most researchers

apply, is the characterization of the model transitions, to mixtures of base densities g of a family G that have a simple parametric form. Densities $g \in G$ are usually Gaussian or Laplacian, and can be parameterized by the mean vector and the covariance matrix [19]. HMMs with these kinds of distributions are usually referred to as *continuous HMMs.* To model more complex distributions, a rather larger number of base densities has to be used in every mixture. This may require a very large training set of data for the estimation of the distribution parameters. Problems arising when the available corpus is not large enough and can be lightened by sharing distributions among transitions of different models. In *semi-continuous HMMs* though, all mixtures are expressed in terms of a common set of base densities. Different mixtures can be characterized only by different weights [19].

The computation of probabilities with discrete models is quicker than with continuous models, nevertheless it is possible to speed up the mixture densities computation by applying *vector quantization* (VQ) on the Gaussians of the mixtures [5]. Finally, the parameters of HMMs can be estimated by iterative learning algorithms [52] in which the likelihood of a set of training data is guaranteed to increase at each step.

4.2.2 The problem with HMMs

Two are the assumptions that characterized the HMMs; firstly, the Markov chain assumption, referring that the current state depends only on the previous state given the current state (in a first-order Markov chain). Secondly, the output independence assumption, in where a particular symbol that is emitted at time t, depends only on the state s_t given this state, and is conditionally independent of the past observations. These assumptions puts however some significant limitations on the accuracy of the

model [12]. It is difficult to model relative durations within a phone segment since the mode sequence in HMMs is a Markov chain. It is possible to have some parts of a segment stretched and others compressed, and this does not agree with speech knowledge. The biggest deficiency of HMMs comes however from the second step and the conditional independence assumption of the observations. The modeling of inter-frame time correlation is done through the statistics of the mode sequence only. In particular if we use continuous output distributions then one can show [45] that the joint likelihood of a particular mode and observation sequence is dominated by the terms of the output distributions as the dimension of the feature vector increases. In other words an HMM with continuous output distributions converges asymptotically to an independent process with the dimension of the feature vector and this we believe is not a realistic model for speech.

4.3 Artificial Neural Networks

Artificial Neural Networks (ANNs), introduced in 1943 by McCulloch and Pitts [44], are variations on the *parallel distributed processing* (PDP) idea. This is why ANNs are also known as *connectionist models* or *parallel distributed processing*. Due to their nature, ANNs are particularly interesting for automatic speech recognition, which requires a series of constraints to be satisfied, i.e., the parallel evaluation of many clues and facts and their interpretation in the light of numerous interrelated constraints [27].

ANNs consist of a number of nodes, or units, connected with each other by links [64]. Each link has a probabilistic weight, and the learning procedure is taking place

by updating these weights with the new estimated ones. There are some units which are connected to the external environment, and these can be consider as the input or output units. Each unit has a set of input links from other units, a set of output links

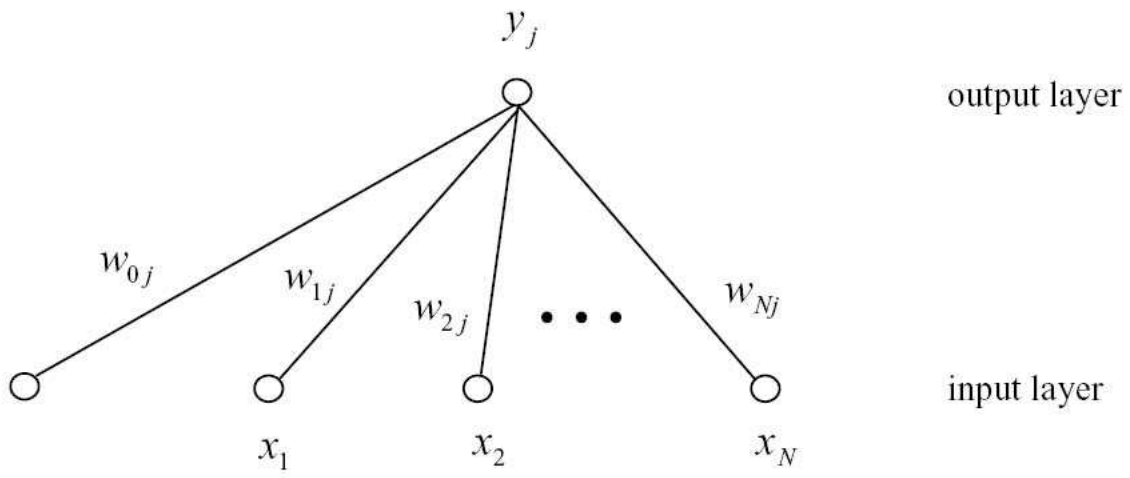

Figure 4.3: A single-layer perceptron

to other units, a current activation level, and a means of computing the activation level at the next step in time, given its inputs and weights. The idea is that each unit does a local computation based on the inputs from its neighbors, independently without any kind of control over the rest units. In practice, due to the fact that most neural network have been implemented as a software program, it is customary to use synchronous control to update all the units in a fixed sequence. To build a neural network to perform an application task, one must first decide how many units are to be used, what kind of units are appropriate, and how the units have to be connected to form a network. One then initializes the weights of the network, and trains the weights using a learning algorithm applied to a set of training examples for the task.

A simple neural network consists of one layer and is called a *single-layer perceptron.*

In figure (4.3) it is shown graphically the structure of a such network [27]. The output y_j, which is non-linear, can be defined as

$$y_j = f\left(w_{0j} + \sum_{i=1}^{N} w_{ij}x_i\right) = \mathbf{w}_j\mathbf{x}^t = g_j(\mathbf{x}) \tag{4.3.1}$$

where $\mathbf{w}_j = (w_{0j}, w_{1j}, w_{2j}, ..., w_{Nj})$ are weights and $\mathbf{x} = (1, x_1, x_2, ..., x_N)$ are the N inputs. Each class ω_j is associated with a discriminant function $g_j(\mathbf{x})$ as shown in (4.3.1). The decision rule can be based on these discriminant functions, and can be given by

$$\mathbf{x} \in \omega_k \Longleftrightarrow k = \arg\max g_i(\mathbf{x}). \tag{4.3.2}$$

A more complex form of ANNs, is the *multi-layer perceptron* [26, 63] shown in figure (4.4) with four layers with the two middle ones being hidden, obtained by [27]. Each layer has the same computation models as the single-layer perceptron.

When one is intending to deal with speech, which is a nonstationary signal, has to take into consideration a few important facts. For instance, he needs to address how to map an input sequence properly to an output sequence when the two sequences are not synchronous. In this consideration, one should include a proper alignment, segmentation, and classification rule. The basic neural networks though, are not good enough to address these problems in a unified way. On the other hand, *recurrent* neural networks have an internal state that is a function of the current input and the previous internal state. One of the popular neural networks is the *Time Delay Neural Network* (TDNN) [68]. The TDNN can be used in a training process, to recognize a sequence of predefined length. The activation in the hidden layer is computed from the current and multiple time-delayed values of the previous layer, and the output

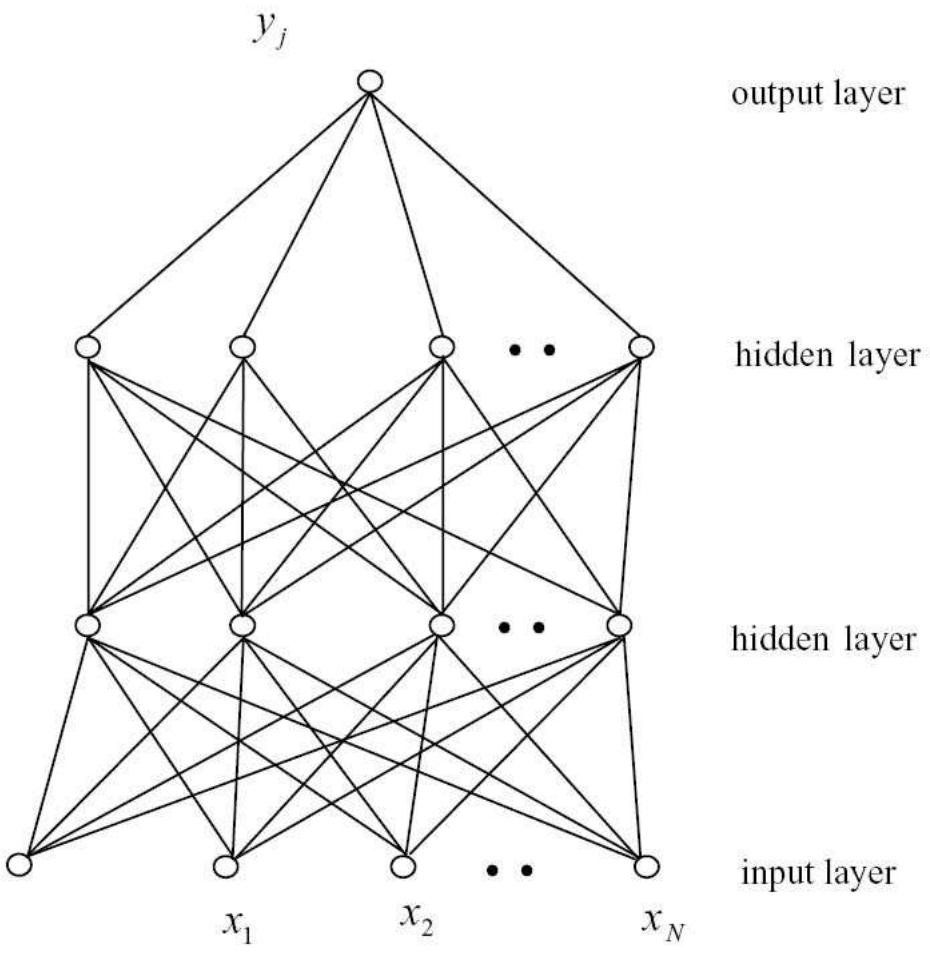

Figure 4.4: A multi-layer perceptron with four total layers. The middle two layers are hidden.

units are activated only when a complete speech segment has been processed. A typical TDNN is illustrated in figure (4.5), which we have also obtained by [27]. The TDNN has been successfully used to classify pre-segmented phonemes.

For small vocabulary speech recognition tasks, ANNs seem to have better performance than HMMs, especially when they have to deal with short, isolated speech units. Nevertheless, it still remains a challenge for researchers that involved with neural networks, to prove in practice that they can be as effective as HMMs for large

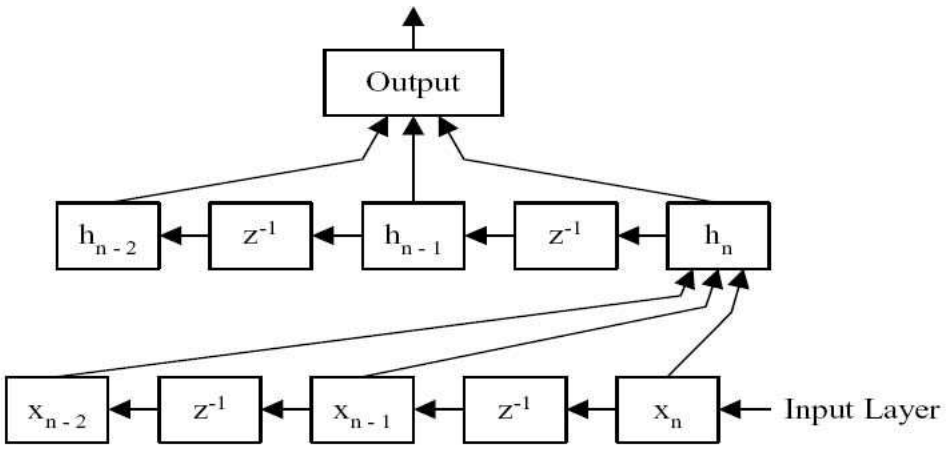

Figure 4.5: A Time Delay Neural Network (TDNN). h denotes hidden, x_t the input vector at time t and z denotes a delay of one sample.

vocabulary speech recognition applications.

4.3.1 Hybrid HMMs/ANNs

In dealing with continuous speech tasks, the most effective solution is to integrate neural networks with HMMs [16, 17]. Instead of the gaussian mixture densities that we have in HMMs, ANNs can be used as the output probabilities replacing the gaussian mixtures. The output probabilities could be estimated in the same way as in standard HMMs, but this time the Bayes rule should be applied to the output of the ANNs that have been trained to classify the HMM state categories. The neural networks can consist either of a single large trained network or of a group of separately trained small networks [9, 21, 46, 59]. A number of techniques have been developed in order to improve the training performance of these networks. Training can be embedded in an EM-style process. For example, dynamic programming can be used to segment the training set and then the segmented data are used to retrain the neural network. In

addition, it is also possible to have a Baum-Welch style training [6, 25] as in Hidden Markov Models.

4.4 Segment Models

Previously, we mentioned that in order to apply HMMs in acoustic modeling, one should make a few assumptions, that are not always corresponding to reality. One of them is the state conditional independence assumption. Instead of generating conditionally independent observation vectors, the discrete states in a segment model generate a sequence of observations length l [48, 60] as shown in figure (4.6), obtained by [48].

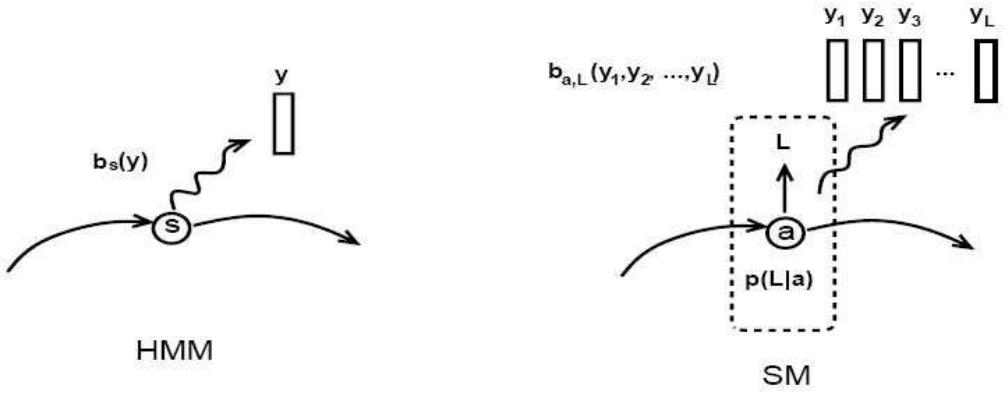

Figure 4.6: A HMM generates one single frame y. On the other hand, SM generates a random length sequence of frames $(y_1, ..., y_l)$.

When we are referring to a segment in speech, we mean a variable-length part of the speech waveform [12], that usually corresponds to a language unit, a word, a phone or a sub-phone. Segment-based models [4, 7, 33, 43, 50, 62, 18] have been proposed as HMMs alternatives, offering a more suitable and flexible scheme to model the

dynamics of speech signal. In [47, 50, 61], a ***Stochastic Segment Model*** **(SSM)** was introduced for continuous speech recognition. In this, one assumes that the observation sequence $Y = [y_1, ..., y_l]$ of random length l, is generated by each segment of speech, and the model for a phone α consists of a family of joint density functions (one for every observation length), and a collection of mappings that specify the particular density function for a given observation length [49]. The segments are described by a fixed length sequence of locally time-invariant regions. To specify which observation vector corresponds to each of these regions, one uses a deterministic mapping.

The probability of a segment given phone α is the product of the probability of each observation y_i and the probability of its duration L, which is known

$$p(Y|\alpha) = p(Y, L|\alpha) = p(L|\alpha) \prod_{i=1}^{L} p(y_i|\alpha, T_L(i)) \tag{4.4.1}$$

where $T_L(i))$ is the corresponded region and determines the mapping of the L-long observation to the m regions in the model.

One of the basic problems with this model, was the training procedure. A first approach was a full-covariance Gaussian SSM, an approach that requires a large number of parameters. However, it takes full account of the intra-segmental correlations within a speech segment. Robust estimation of such model would require a great amount of data, and in practice was found to be inappropriate for systems having larger feature dimensions. For this reason, the idea of using a block-diagonal covariance was adopted. The basic assumption was that successive frames are independent given the length l of the segment. Even though the inter-frame correlation modeling - the potential advantage of SSMs - is ignored under a such model, it has been shown to match the performance of HMMs.

In [12], a Gauss-Markov SSM was introduced to solve the above problems. Thus, the model that was looking for should have a moderate number of free parameters without trading its capability to model time correlation efficiently. The density of the unobserved feature sequence y given phone α is characterized by a Markov assumption

$$p(\mathbf{y}|\alpha) = p(y_1|\alpha)p(y_2|y_1,\alpha)p(y_3|y_2,\alpha)...p(y_M|y_{M-1},\alpha) \tag{4.4.2}$$

where $p(y_k|y_{k-1},\alpha)$ are the conditional Gaussian densities of the jointly normal random vectors y_k, y_{k-1}. This model has the advantage that the number of parameters being estimated, increases by less than a factor of 3 over the block-diagonal case, whereas it still models time correlation. Experiments shown that there was no significant improvement in performance over the independent-frame model (diagonal-covariance SSM). Due to the nature of speech, it is quite possible to have large differences between speakers, dialects, and recordings. Hence, they thought that they can get significant improvements if they could foresee possible mismatches and smooth the estimated distributions. The idea was to achieve a more smooth version of Gauss-Markov model, by adding an observation noise component to the model. Thus, a linear state-space dynamic system or ***Linear Dynamic Segment Model*** **(LDSM)** was constructed. The set of equations for the state and the observation space respectively was then

$$x_{k+1} = F_k(\alpha)x_k + w_k \tag{4.4.3}$$

$$y_k = H_k(\alpha)x_k + \mu_k(\alpha) + v_k \tag{4.4.4}$$

where w_k, v_k were uncorrelated zero-mean Gaussian vectors with covariances $P_k(\alpha), R_k(\alpha)$ respectively. In the above, the structure of the model was such, in order to ensure that the system was identifiable [8, 35].

4.4.1 Other segment-based models

In previous years, another approach, similar to SSMs, has been proposed in bibliography called ***Hidden Dynamic Models*** **(HDMs)** [11, 42, 51, 55, 70]. In this, one tries to better model the co-articulation phenomenon and the transitions between neighboring phones. The HDM consists of a single vector target per phone in a hidden dynamic space in which the trajectories of speech are produced by a dynamic system. The observation process in HDMs is implemented by a global *multi-layer perceptron* (MLP). The model has a simple, and a very flexible structure and it can capture important aspects of the relationship between phonetic labels and acoustic patterns. However, the inference algorithms for the HDM are not really tractable. A number of approximate methods have been proposed [34, 39, 40, 41, 65].

Another approach in segment-based modeling, was the idea of inserting articulatory knowledge into acoustic models. In [56, 57, 58] the ***Hidden Articulatory Markov Model*** **(HAMM)** was introduced which directly integrates articulatory information into speech recognition. Based on the [15], HAMM essentially is a HMM in which each articulatory configuration is modeled by a separate state. The state transitions aim to naturally reflect human articulation; static constraints disallow configurations which would not occur in a human language such as English, and dynamic constraints ensure that only physically possible movements are allowed. Furthermore, asynchronous articulator movement is allowed as each feature can change value independently of the others. In addition to the static constraints which reduced the number of states from 25600 to 6676, the number of parameters was further reduced by removing states with low occupancy during training process. The recognition task

was the PHONEBOOK, an isolated word, telephone speech corpus. With a 600 word lexicon, the HAMM gave a significantly higher word error rate than a standard 4-state HMM. However, a combination of the models gave a word error rate smaller than the HMM.

In [20], a ***Linear Dynamic Model*** **(LDM)** is proposed. LDM is based on the work of Digalakis et. al., mentioned previously, and examines the assumptions made in applying such a model in speech recognition and shows that the addition of a hidden dynamic state leads to increases in accuracy over the equivalent static models. LDMs are suitable in modeling smoothly varying, continuous, noisy trajectories such as found in measured articulatory data. In this work, it was also introduced a novel approach of decoding for segment models in the form of a stack decoder with A^* search. Such a scheme allows flexibility in the choice of the acoustic and language models since the Viterbi criterion is not integral to the search, and hypothesis generation is independent of the certain language model. Furthermore, the time-asynchronous ordering of the search means that only the most likely paths are extended, and so a minimum number of models are evaluated.

Finally, in [60], there were examined several state-space modeling approaches. The state evolution and observation processes are assumed to be linear and noise sources are distributed according to Gaussians or Gaussian mixture models. The ***Factor Analyzed HMM*** **(FAHMM)** was the first model to be investigated in the referred work. In this, the state evolution process is assumed to be piece-wise constant. All the variations of the state vector about these constant values are modeled as noise. In the FAHMM a discrete Markov random variable chooses the continuous state and the observation process parameters. The FAHMM generalizes a number of

standard models such as the independent factor analysis, shared factor analysis and semi-tied covariance matrix HMMs. A second approach examined in the same thesis [60], was the case that the state evolution process is assumed to be a linear first-order Gauss-Markov random process. Using Gaussian distributed noise sources and a factor analysis observation process this model corresponds to a LDS. For acoustic modeling, a discrete Markov random variable is required to choose the parameters of the LDS. This hybrid model is called the ***Switching Linear Dynamical System* (SLDS)**. The SLDS is related to the stochastic segment model, which assumes that the segments are independent. In contrast, for the SLDS the continuous state vector is propagated over the segment boundaries, and hence provides a better model for co-articulation. Unfortunately, the inference for the SLDS is intractable due to the exponential growth of posterior components as time passes. In the experiments was found that a FAHMM system with very basic configuration could outperform or achieve an equal performance to a standard diagonal covariance matrix HMM with fewer parameters. The SLDS systems generally were outperformed by the SSM systems, which were outperformed by the baseline FAHMM.

4.5 Linear Dynamic System Segment Models

Our proposed General Identifiable State-Space Models (GISSMs) could be applied to speech recognition, as an acoustic model. In this case the model is referred as Linear Dynamic System Segment Models (LDSSMs). In the next, we shall present the application process, beginning with the preliminaries, the dataset and the feature selection and extraction processes.

4.5.1 Database

The source speech of AURORA 2 database is the TIDigits consisting of connected digits task spoken by American English talkers (down-sampled to 8 kHz). A selection of 8 different real-world noises have been added to the speech over a range of signal to noise ratios with controlled filtering of the speech and noise. Noise signals are selected to represent the most probable application scenarios for telecommunication terminals. Noises have been recorded at different place s: 1) Suburban train, 2) Crowd of people (babble), 3) Car, 4) Exhibition hall, 5) Restaurant, 6) Street, 7) Airport, 8) Train station. The noise signals are added to the TIDigits at SNRs of 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB.

Two training modes are defined as: training on clean data only, and training on clean and noisy (multi-condition) data. Each of the above modes contain 8440 utterances selected from the training part of the TIDigits. For the second mode, the data are equally split into 20 subsets with 422 utterances in each subset. Each subset contains a few utterances of all training speakers. The 20 subsets represent 4 different noise scenarios at 5 different SNRs. The 4 noises are suburban train, babble, car and exhibition hall. The SNRs are 20 dB, 15 dB, 10 dB, 5 dB and the clean condition. Three different test sets are defined. 4004 utterances from 52 male and 52 female speakers in the TIDigits test part are split into 4 subsets with 1001 utterances in each. A noise signal is added to each subset of 1001 utterances at SNRs of 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB. Furthermore the clean case without adding noise is taken as seventh condition. In the first test set, called **test set A**, the four noises suburban train, babble, car and exhibition hall are added to the 4 subsets. The second test set, called **test set B**, is created in exactly the same way, but using the

four different noises, namely restaurant, street, airport and train station. The third test set, called **test set C**, contain s 2 of the 4 subsets with 1001 utterances in each. Again the clean case without additive noise is considered as seventh condition. This set is intended to show the influence on recognition performance when a different frequency characteristic is present at the input of the recognizer.

4.5.2 Front-End

Although all HTK tools can parameterize waveforms on-the-fly, in practice it is usually better to parameterize the data just once [69]. The tool HCopy is used for this. As the name suggests, HCopy is used to copy one or more source files to an output file. Normally, HCopy copies the whole file, but a variety of mechanisms are provided f or extracting segments of files and concatenating files. By setting the appropriate configuration variables, all input files can be converted to parametric form as they ar e read-in. Thus, simply copying each file in this manner performs the required encoding.

HTK support both FFT-based and LPC-based analysis. In our experiments we used Mel Frequency Cepstral Coefficients (MFCCs), which are derived from FFT-based log spectra. Coding can be performed using the tool HCopy configured to automatically convert its input into MFCC vectors. A configuration file (config) is needed which specifies all of the conversion parameters.

Our settings for these are as follows, even though some of them are in fact the default setting. In brief, they specify that the target parameters are to be MFCC using C_0 and energy E, data were created on a big-endian SUN machine, the source file has no header, the frame period is 10msec (HTK uses units of 100ns), the output

should not be saved in compressed format, and no crc checksum should be added. The FFT should use a Hamming window and the signal should have first order pre-emphasis applied using a coefficient of 0.97. The filterbank should have 23 channels and 12 MFCC coefficients should be output. The sample rate is set to be at 1250 since data were down-sampled at 8 kHz. Lower and higher frequencies cut-offs are set at 64 and 4000 respectively.

4.5.3 Learning LDSSMs

We then investigated the performance of the linear dynamic system on a classification problem on real speech data. We chose the connected digits task of the clean Aurora 2 database for our experiments and trained our models using 4000 training sentences. We train a total of 11 word-models, 9 for the digits *one* to *nine* and 2 for the *zero* and *oh* words. Each word model consists of several segments each of which is modeled with a linear state-space model as the one defined in equations (3.1.1), (3.1.2) or (3.2.3). The number of segments is defined according to the number of phonemes of each word as it shown in the following table. Dummy silence "sil" and short pause "sp" models were also considered for implementation issues.

Words	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Zero	Oh
Segments	6	4	6	6	6	4	8	4	6	6	2

Table 4.1: Number of Segments for each Aurora 2 Word-Model.

The modeling of a 4-segment word-model and the corresponding state-space equations are shown in the example of figure 4.7. It can be seen that the correlations between segments are modeled through the time transitions of the linear systems.

Based on this configuration, it is obvious that the training of the word-models requires frame-segment alignments. Such alignments can be ideally obtained through dynamic programming. However, we choose to simplify our implementation at this phase and obtain these alignments using HTK and equivalent, well trained HMMs consisting of as many states as the number of segments defined in each of the word-models.

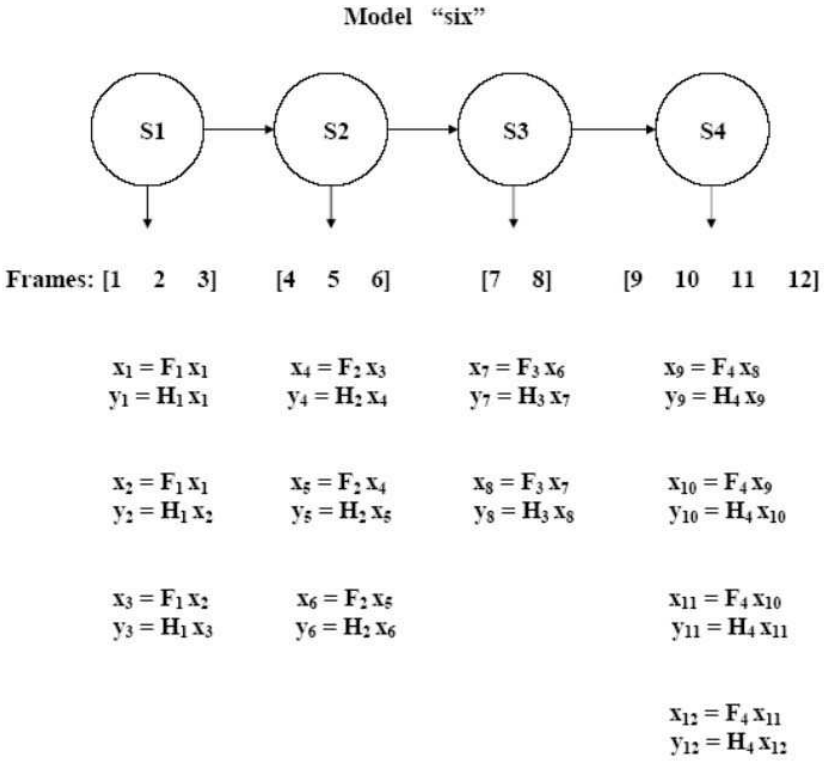

Figure 4.7: Training scheme for a 4-segment word-model.

We choose the dimension of the state to be equal to the size of the observation vector which means that the F matrix is filled with free parameters and the observation matrix is set to the identity matrix $H = I$. The initial values of the noise covariances P and R are set randomly while the initial state-transition matrices F are estimated

from the training data. The forward and backward runs are then applied to the whole training sentence.

4.5.4 Recognition

For the classification we choose single-word test utterances that were not used for training. The test set consisted of 600 isolated words distributed uniformly among the word-models. We obtain alignments for each of the test sentences and each of the word-models and we sum the log-likelihoods obtained from the forward run on each trained word-model. We finally classify the sentence to the word that achieved the highest likelihood.

The learning and the recognition part of the LDSSMs, was done by Georgios Tsontzos and Vassilios Diakoloukas. In these two tasks of our speech recognition application, the author had a more auxiliary and advisory role. The feature extraction and the frame-segment alignments though, was carried out by the author.

4.6 Experimental Results

Figure 4.8 shows classification performance (% word-accuracy) for models trained using three different initialization sets for the noise covariances P and R. Figure 4.9 presents the increase of the log-likelihood for the same training runs. As can be seen from both diagrams there is fast initial convergence of the EM algorithm and the best performance is achieved after only a few EM iterations. However, results also indicate strong relation of the initialization of the noise covariances P and R and the performance of the final system.

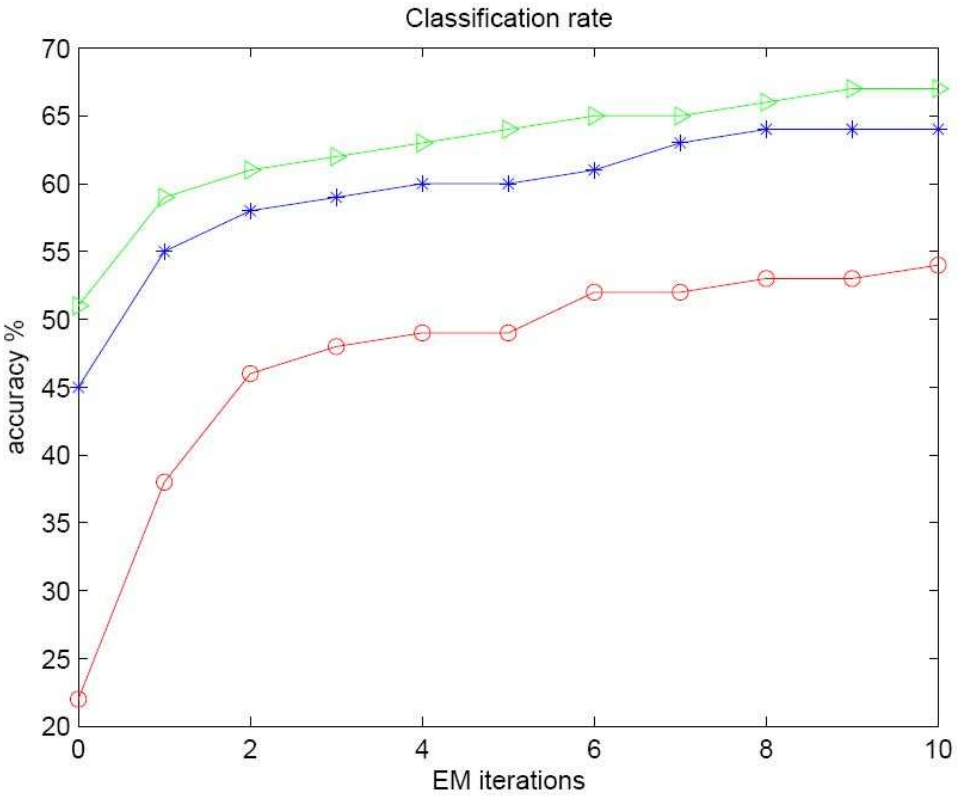

Figure 4.8: Classification performance of test data vs. number of EM iterations for models trained with three different initialization sets.

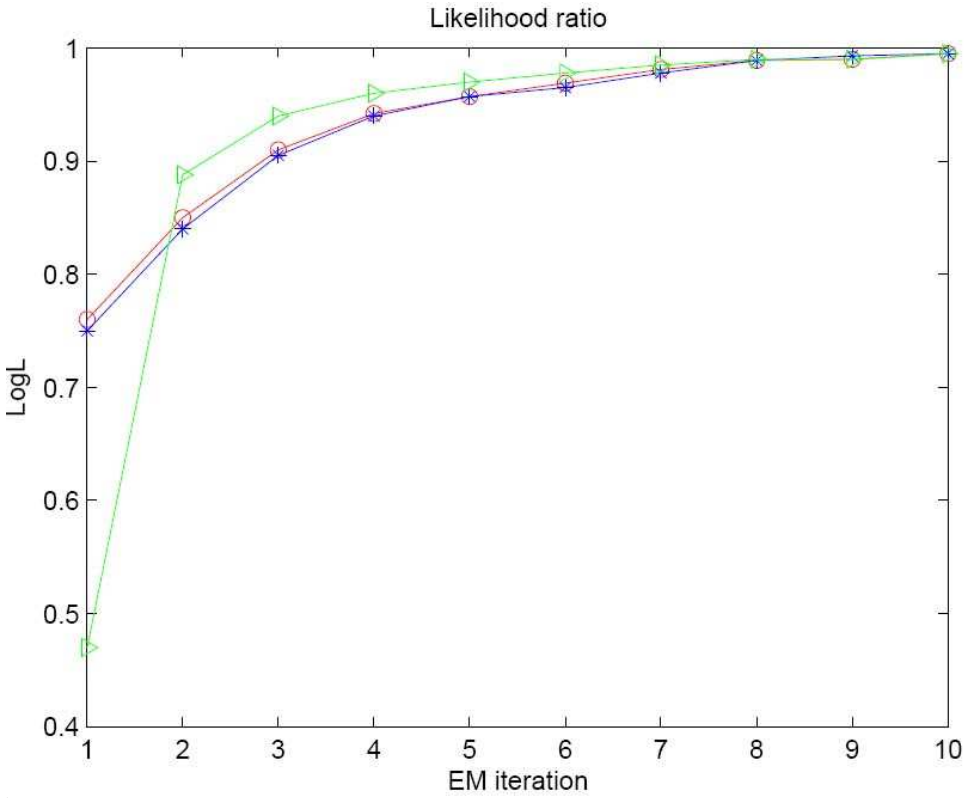

Figure 4.9: Normalized log-likelihood ratio of each iteration relative to the convergent value for the training data, for models trained with three different initialization sets.

4.7 Discussion

Stochastic processes are a standard in acoustic modeling. We have a small review of the most popular statistical approaches. Linear State-Space Models could be a promising solution for speech recognition, since they can visualize the mechanism of speech production system in a more accurate form. The development of such models is a very difficult procedure and requires patience and hard work.

We performed training and classification on speech data based on speech segments and the generalized identification state-space models equations. The results show significant correlation of the system performance and the initialization of several system parameters.

A set of improved experimental results will be presented in the near future, within a diploma thesis written by Georgios Tsontzos.

Chapter 5

A Weather Application: three-days forecasting

The basis of a science is its ability to predict.

R. P. Feynman

As we have already mentioned from the beginning of this dissertation, our goal was to develop such a linear dynamical system, so as to be easily applied in various tasks, without constraints and limitations. In this chapter we show how the proposed system could be applied in a weather prediction problem. The task is simple; to predict the temperatures over the next three days from a few weather observations.

5.1 Database

The database is a part of author's personal file, retrieved from the internet during the last three years. Data concerning the city of Chania (Souda airport) and were collected from the yahoo's weather service at the following URL: `http://weather.yahoo.com/forecast/GRXX0032_c.html?force_units=1`. We considered temperatures at three different times per day; the first measurement was taken at 6.50 am

referred as the morning measurement, the second at 3.50 pm considered as the midday measurement and the third at 11.50 pm referred as the night measurement.

Months from January to June contain data from three years $(2004, 2005, 2006)$. On the other hand, months from July to December only contain data from years 2004 and 2005 (for practical reasons). For our experiments, we used data only from January to June. Two were the main reasons for considering this decision; firstly because of the symmetry in temperatures between the first and the second half of a year that could probably cause some ambiguity to the system (spring and autumn have quite the same temperatures and the same is for July and August), and secondly because for the current year 2006, we do not have temperatures over the second half of the year.

For each month, we divided the available data into training and testing subsets. We used data from years 2004 and 2005 as training data and we kept data from 2006 as testing. Training data were saved in six different files, each of which was used for estimating the models parameters. On the other hand, testing data were saved all in one file; rows from 1 to 31 corresponded to January 2006, from 32 to 59 corresponded to February 2006, from 60 to 90 to March 2006, 91 to 120 to April 2006, 121 to 151 to May 2006, and finally rows from 152 to 181 corresponded to June 2006.

5.2 Training the month-models

The first step was to train the system by inserting the training observations, so as to extract 6 models, one for each of the months from January to June. Starting from January, we inserted the data observations into the system, considered to be

the observation vector y_k. At the beginning we randomly initialized the systems parameters F, P and R as well as the initial values of the covariance Σ_x of the state. For the initial mean value μ_x of the state vector, we set it to be equal to zero. Training was carried out by inserting the data within a three-day segment. This means that the state vector was a 9×1 vector and the dimension of the parameters matrices $F, H, P, R,$ was 9×9. In essence, the observation vector consisted of a set of three days. Consider for example three continuous days. The first day has three samples $[a_1 b_1 c_1]^T$ for the morning, midday and night measurement. The second day consists of the $[a_2 \ b_2 \ c_2]^T$ and the third of $[a_3 \ b_3 \ c_3]^T$. In this case the first three-day segment is $[a_1 \ b_1 \ c_1 \ a_2 \ b_2 \ c_2 \ a_3 \ b_3 \ c_3]^T$. The second three-day segment is $[a_2 \ b_2 \ c_2 \ a_3 \ b_3 \ c_3 \ a_4 \ b_4 \ c_4]^T$ where the index 4 refers to the upcoming day. Hence, every day can be found within its neighboring days. In this way, we give some extra information to the model about the behavior of the temperature across the time. The training process followed the same path as in our previous experiments. The training data passed through the Forward and Backward recursions of the Kalman smoother and then the sufficient statistics were computed. The new estimates of the month-model parameters, obtained from these statistics according to the element-wise estimation equations given in (3.3.2) - (3.3.4).

5.3 Forecast

We then used the trained month-models to perform a three-day weather forecasting or, to be more precise, a three-day temperature forecasting since we only try to predict the temperatures and not other weather conditions such as humidity, winds, clouds, rain, sun, snow etc.

At the beginning, the system was initialized according to the parameters of the model January. Next, we inserted the testing data, recalling that the first 31 samples corresponded to January, also in a three-day segment as in the training process. In every iteration of the Forward algorithm, and before continuing with the next sample, we also computed the $y_{k+1|k}, y_{k+2|k}$ and $y_{k+3|k}$ which are our estimates over the next three time steps. Essentially, these values are our temperature prediction of the following three days-segments, based on the temperatures of the current day-segment, namely the $y_{k|k}$ or simple y_k, following the notation of the Forward recursions presented in chapter 3. We followed the same procedure for all the testing data. The parameter set automatically switched to the values of the February model, when the first day of a three-day segment was a February sample. Every time the first sample of the next segment was from a different month, the system immediately turned to the corresponded model. In the end, for every day we obtained a three-day prediction.

5.4 Experimental Results

In this section we shall present interesting experimental results extracted from our application in weather forecasting. Figures (5.1), (5.2), and (5.3) shown norm 1 distances between the real temperatures and the predicted ones from GISSM for the next day, after two days and after three days respectively. We used the same distance metric as we did in our experiments with artificial data shown in equation (3.4.1). Figures shown that the error is larger for the second day of prognoses compared to the first, and even larger for the third day compared to the second, as we normally expected.

In order to compare our results, we did the next experiment. We assumed on each

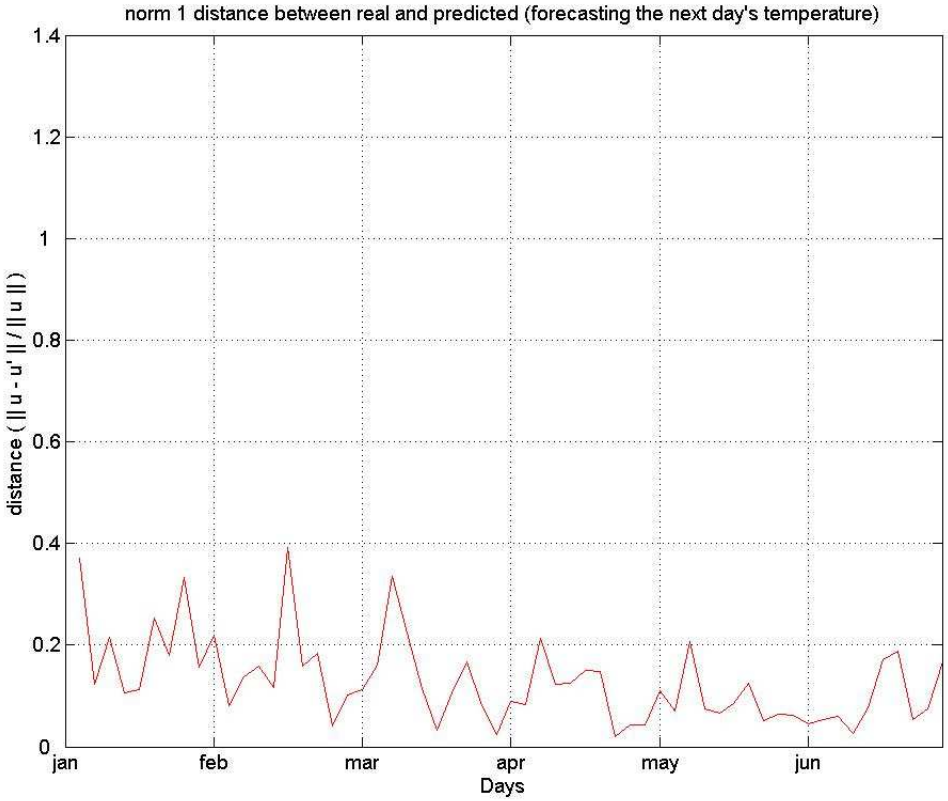

Figure 5.1: Distances between the actual and the predicted temperatures considering temperatures forecasting after one day.

day that the following three days will have the same temperature as the current day. In other words, we simulated a "naive weather predictor" defending that temperatures would be the same for a time period of three days. Figures (5.4), (5.5), and (5.6) shown norm 1 distances between the actual temperatures and the "naive prognoses" for the next day, after two days, and after three days respectively. Studying these figures, one can easily conclude that the GISSMs' prognoses is significant better than the "naive predictor". GISSM predictor clearly outperforms the "naive predictor" during winter time, which is the most flexible season concerning temperatures. "Naive predictor"

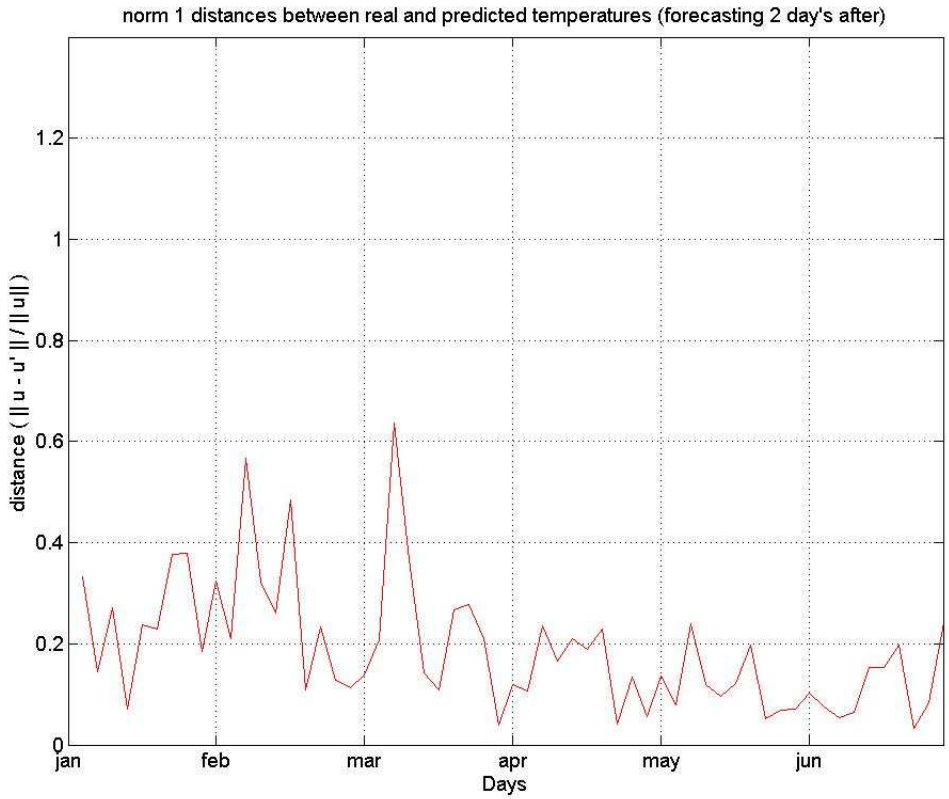

Figure 5.2: Distances between the actual and the predicted temperatures considering temperatures forecasting after two days.

cannot predict the weather dynamical characteristics. The performance of the two predictors for summer time (May and June), was pretty the same mostly due to the fact that in this time of the year, weather in Chania is quite the same every day with small temperature divergences in a time window of three days.

In the next, we show absolute values of the mean temperature error for the three-day prediction with the GISSM predictor (figure (5.7)). Figure (5.8) show its standard deviation. In figures (5.9) and (5.10), we present the same as above for the case of the "naive predictor".

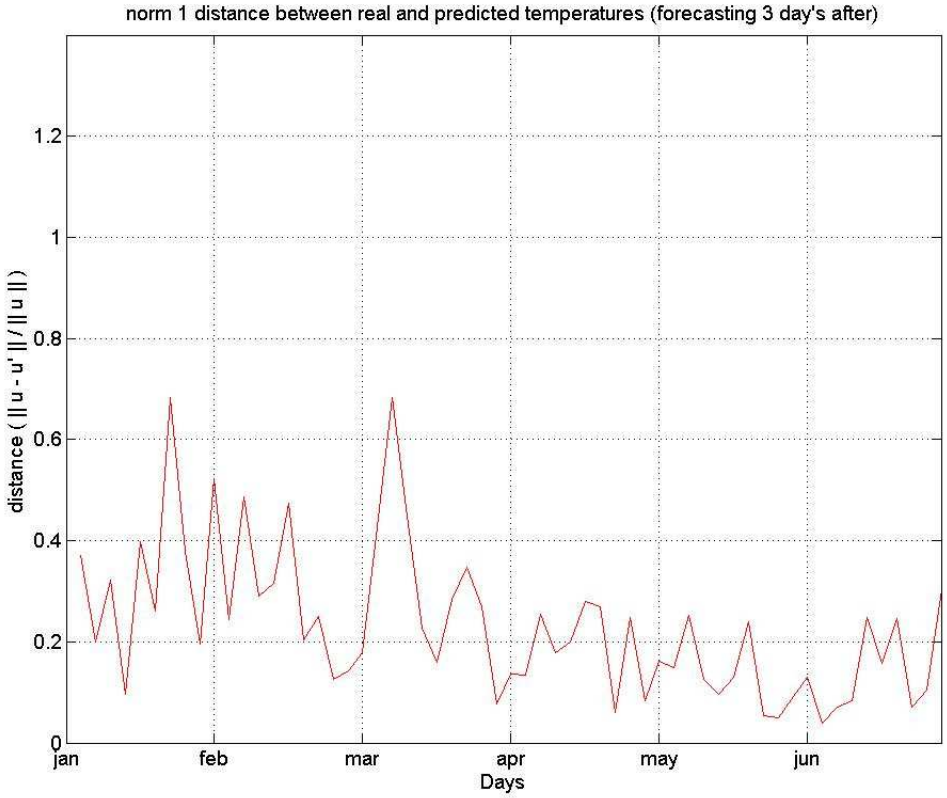

Figure 5.3: Distances between the actual and the predicted temperatures considering temperatures forecasting after three days.

Finally, we compare the two mean error values of the two predictors, for the first, second and the third day (figures (5.11), (5.12), and (5.13) respectively). In these, one can clearly notice the undoubted domination of GISSM predictor. In table (5.1) it is shown the mean temperature error and its standard deviation for the two predictors.

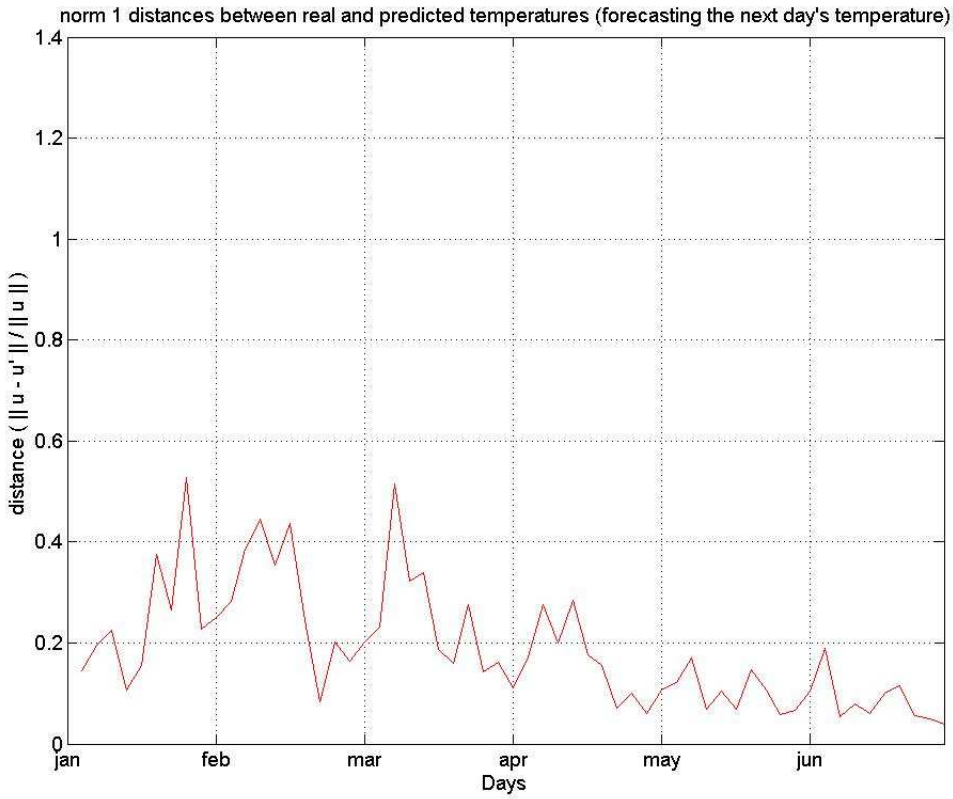

Figure 5.4: Distances between the actual and the predicted temperatures considering temperatures after one day to be the same as the current's day.

forecast	GISSM predictor		"naive predictor"	
	mean temp. error	std	mean temp. error	std
1st day	[1.2362 1.6094 1.4356]	[1.1500 1.4923 1.3451]	[2.0057 2.0961 1.8757]	[1.8768 1.8588 1.9309]
2nd day	[1.8620 2.3252 2.1556]	[1.6557 2.0707 1.9071]	[2.4859 2.9548 2.5028]	[2.1175 2.6845 2.3467]
3rd day	[2.5239 3.1890 2.7741]	[2.0959 2.5913 2.2916]	[2.6045 3.3277 2.7684]	[2.0945 2.9254 2.3636]

Table 5.1: Mean temperature error and its standard deviation for the two predictors.

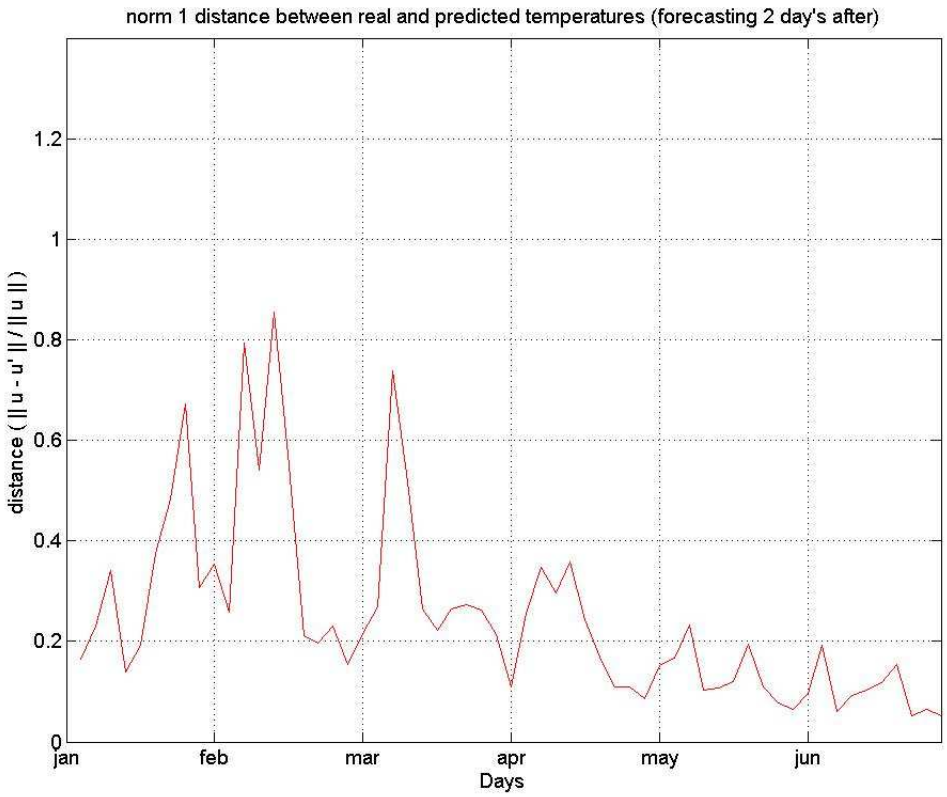

Figure 5.5: Distances between the actual and the predicted temperatures considering temperatures after two days to be the same as the current's day.

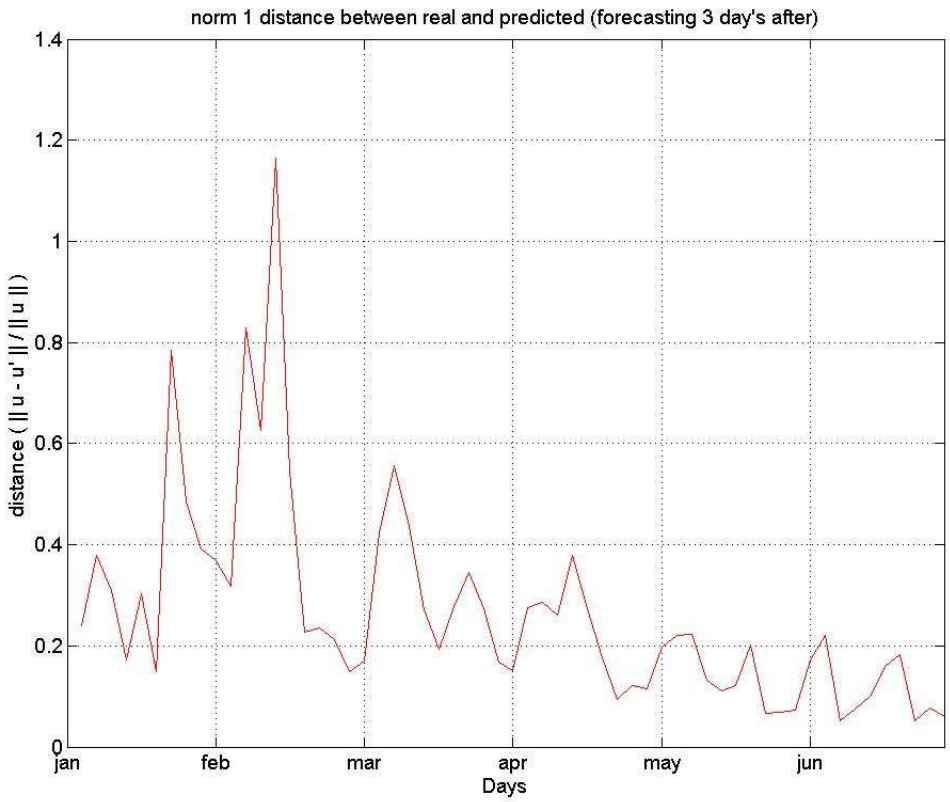

Figure 5.6: Distances between the actual and the predicted temperatures considering temperatures after three days to be the same as the current's day.

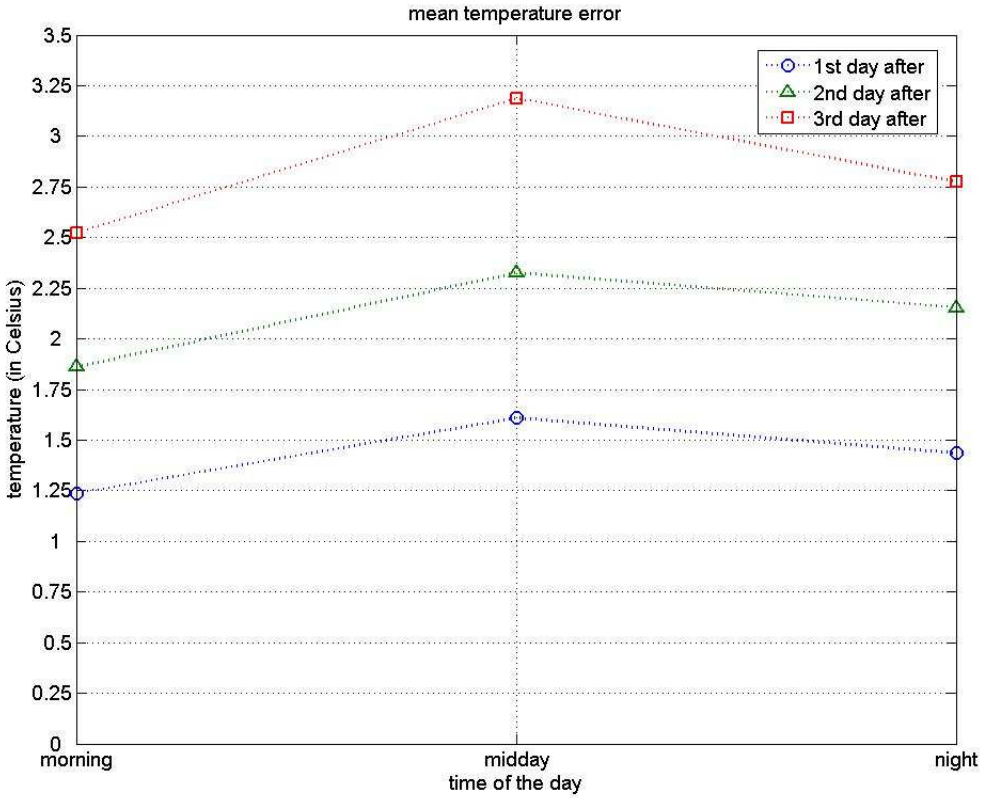

Figure 5.7: Mean temperature error between the actual and the predicted temperatures, using GISSM.

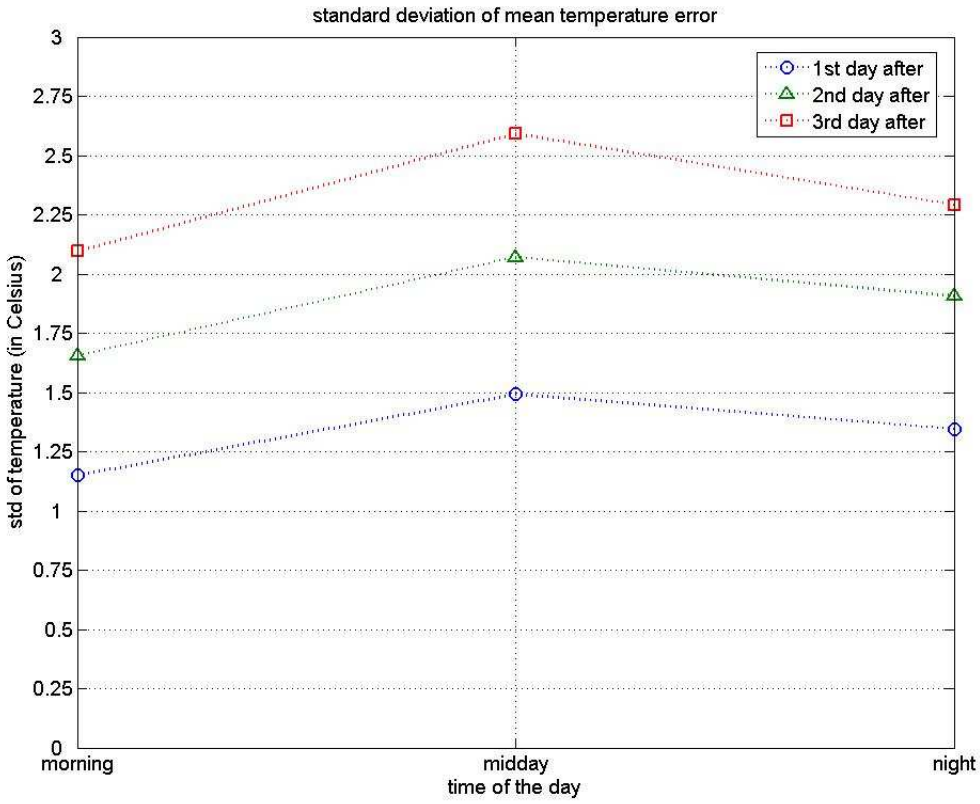

Figure 5.8: Standard deviation of the mean temperature error between the actual and the predicted temperatures, using GISSM.

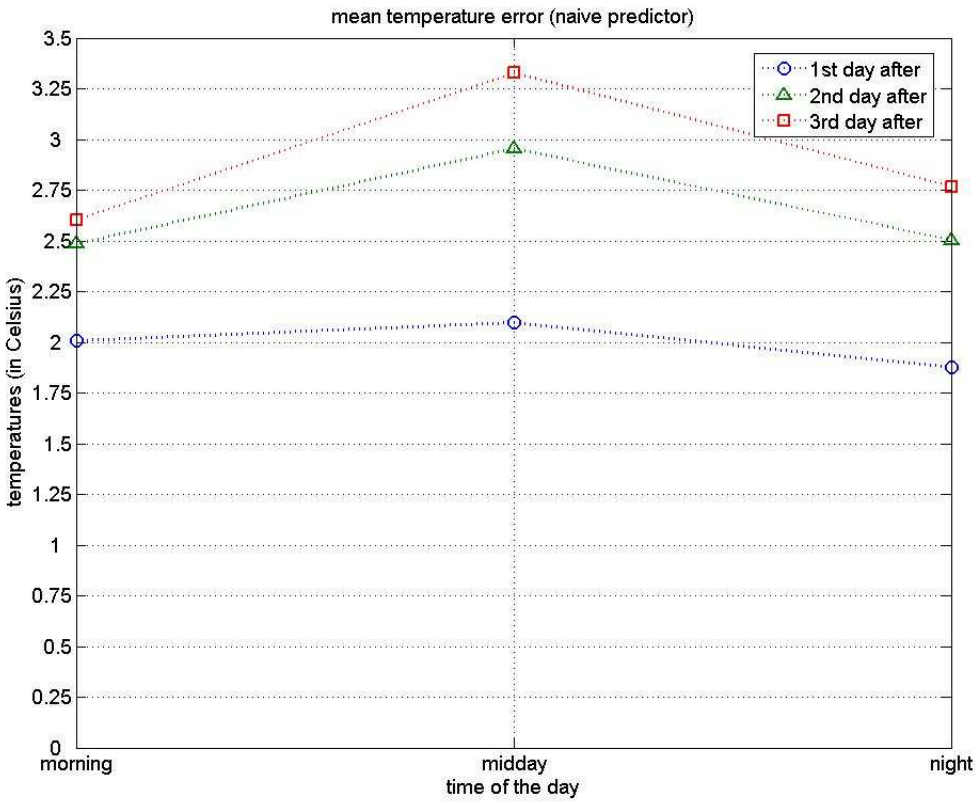

Figure 5.9: Mean temperature error between the actual and the predicted temperatures, assuming the same temperature over the next three days.

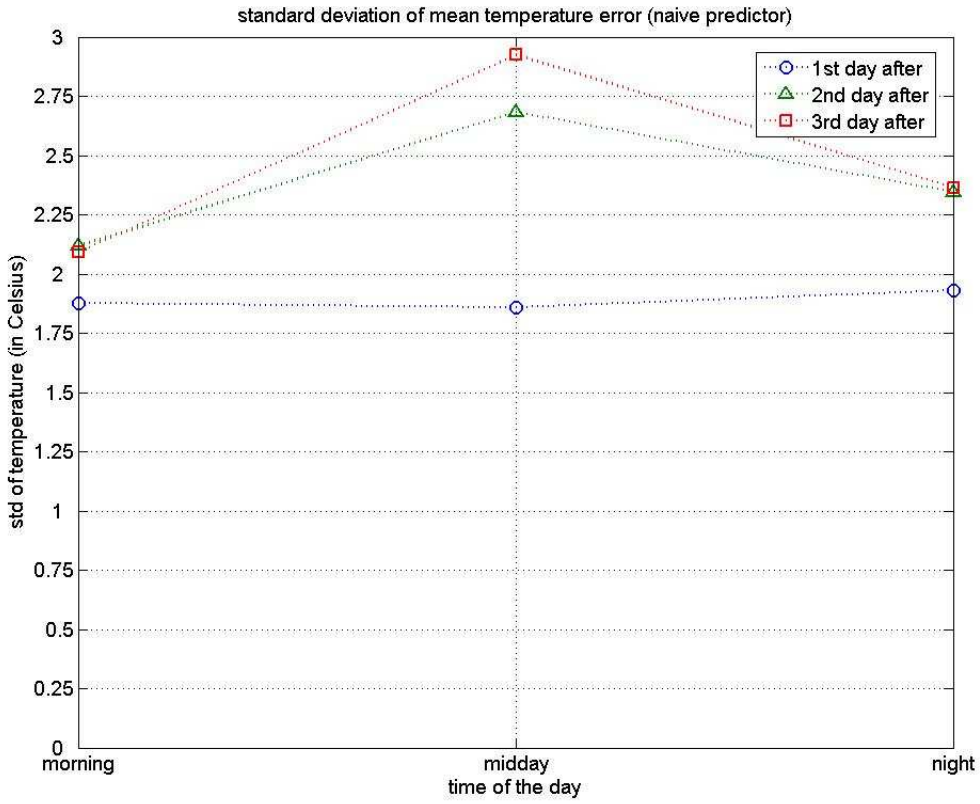

Figure 5.10: Standard deviation of the mean temperature error between the actual and the predicted temperatures, assuming the same temperature over the next three days.

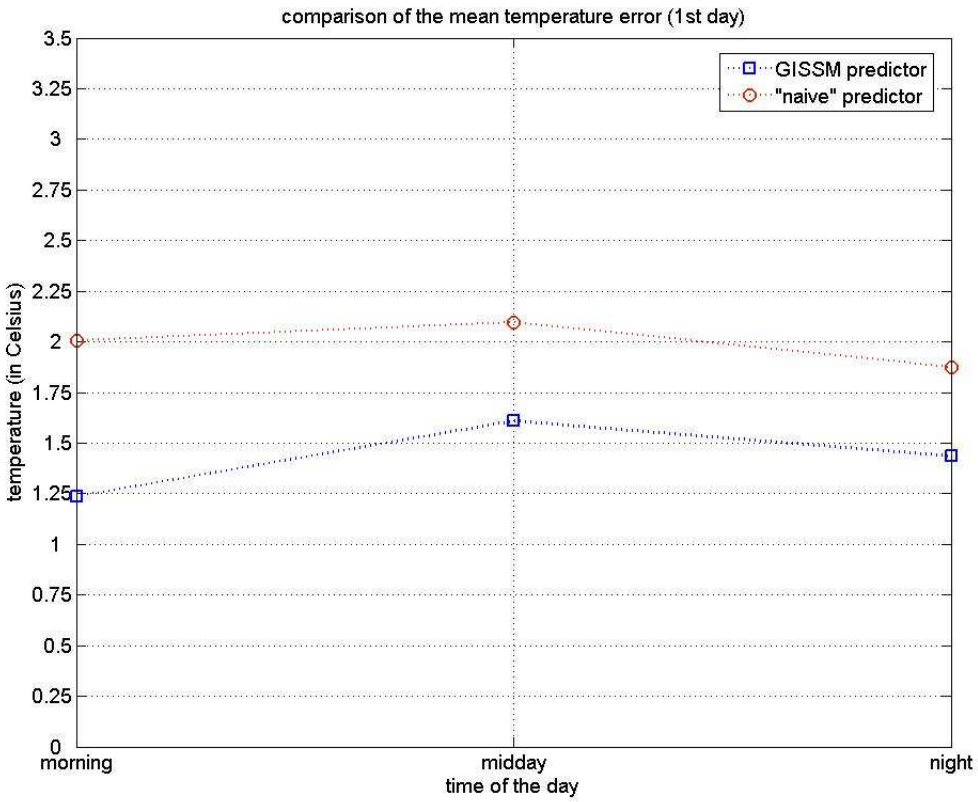

Figure 5.11: Comparison of the mean temperature error in case of prediction with GISSM and when assuming the same temperature. Case of one day after.

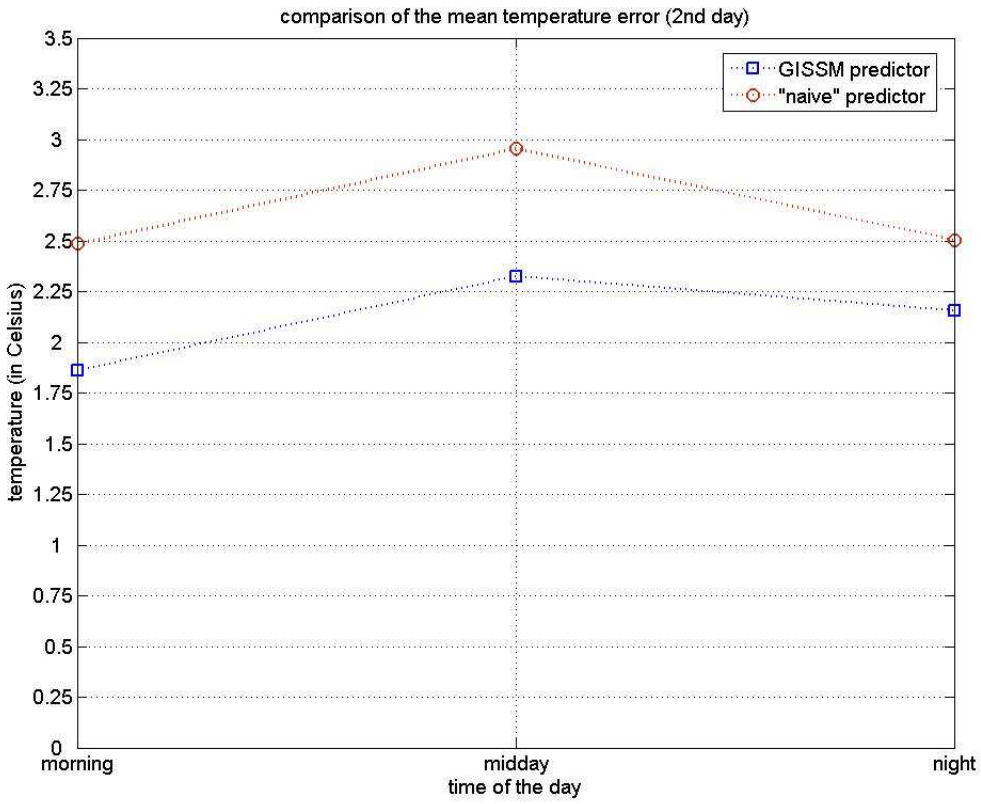

Figure 5.12: Comparison of the mean temperature error in case of prediction with GISSM and when assuming the same temperature. Case of two days after.

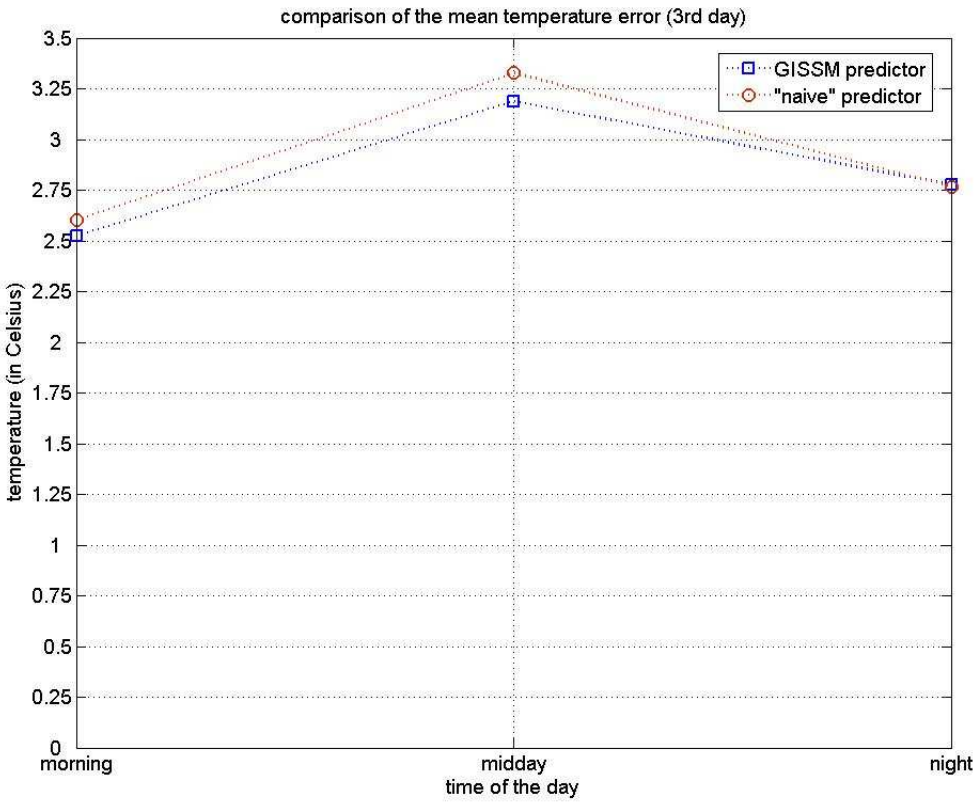

Figure 5.13: Comparison of the mean temperature error in case of prediction with GISSM and when assuming the same temperature. Case of three days after.

5.5 Discussion

We have shown how our system could be used for various applications beyond speech recognition. Essentially, this was achieved without any kind of modification into the structure of the model or into the estimation procedure. The powerful linear dynamic system seems to be able to be applied in a classification/recognition task and also in a prediction/prognoses task, in a very efficient way.

Our experiments in weather prognoses shown that it can achieve good results. Generally speaking, and taking also under consideration the fact that we only had a few temperatures data for training and even less for testing, the behavior of the predictor was effectual. We all know that weather is essentially a non-linear, chaotic phenomenon which can be dramatically affected by many unpredictable and unexpected conditions. Temperature, in particular, could be affected from the physical environment such as sea, mountain chains, forests, etc. or from high or low pressure or from the wind i.e., if it blows from the cold north or from the hot south. However, the GISSM predictor performed well, and in all cases the prognoses that obtained as its outcome was remarkably close to the reality.

Chapter 6

Conclusions

I know nothing except the fact of my ignorance.

Socrates

The goal of this thesis was to investigate new strategies of estimating the parameters of a linear dynamic system and to apply the proposed model to improve the acoustic model in a speech recognition system. We show that our work can be applied beyond the area of speech recognition, by applying it in a weather prediction task. That means that the proposed model could be used in many problems in fields such as estimation, pattern classification/recognition or in prediction problems. In the next section, we first summarize the contributions of this work, which are both theoretical and experimental, and then we highlight the many alternatives that it offers for extension.

6.1 Thesis Contributions

Chapter 2, gives a brief review on Linear Dynamic Models. We presented the main structure of such models that could be found in Dynamic systems literature. We

presented the main tasks that these models could be applied in, depending on whether we performing filtering or learning, and show the different approaches on estimating the parameters and the states of each model.

In chapter 3, we presented generalized forms of linear state-space dynamic models that we were investigated. At the beginning of the chapter, we presented the structure of the proposed model that its canonical form, ensures its identifiability. In our model we overcome the problem with the diagonal noise covariances allowing instead the use of full noise covariances. Moreover the state vectors can arbitrary increased dimension compared to the size of the observation vector. We also developed the training algorithm for this model based on the a non-traditional approach proposed in [12], which is based on the EM algorithm. We extended the proposed training method to an element-wise process and we showed how the equations and the estimation algorithm changes in this way. Next, we investigated the use of an extra control input in the state equation, and we showed how the novel element-wise maximum likelihood approach can be modified in this case. In the end of this chapter, we presented some experimental results with artificially-generated data and studied the convergence of the algorithm.

In chapter 4 we presented the different approaches to the problem of acoustic modeling. We presented the most common methods starting with the most popular, the HMMs. We saw their structure, their mathematical formulation and the applications in speech recognition. We mentioned that the output-independence assumption inherent in HMM modeling is not valid. We also noticed that HMMs cannot model real speech efficiently due to their disability to model more complex dependencies i.e., coarticulation, a very common phenomenon that happens often because articulators,

due to their inertia, cannot make instantaneous transitions from one configuration to the next. Then, we showed neural networks, an approach that, nevertheless is not so famous among speech researches, it can be consider quite efficient, especially for small vocabulary tasks. In the end of this section, we saw some of the work that has been done in the last few years in segment-based models. Since the work that has been presented in this thesis, has a strong relation with these models, we presented a deep review about them. We know that segment models can represent higher order phenomena and the evolution of speech dynamics. However not all of the proposed models have these kind of properties. We mainly focused on linear dynamic models and discuss about the restrictions and constraints that are related with most of them, and the problems that are involved with their application in speech recognition. In the end of the chapter, we represented the application of our GISSM model, which in speech recognition terminology is called LDSSM, in a real speech recognition task. We have shown all the necessary modifications in order to be applied in speech data, and we presented results about the classification accuracy of the system.

Finally, in chapter 5 we have shown in practice how GISSM could be useful in a totaly different task. We have applied our model in a weather forecasting problem trying to perform a three-day temperature prediction, based on our model and a few weather observations. Even though we know that weather prognoses is a difficult non-linear task, results shown that under some circumstances, the General Identifiable State-Space model could be efficiently applied in weather prediction.

6.2 Future Work

There are many suggestions for continuing the work that presented in this dissertation. Basically the extensions are referring to the two applications; speech recognition and weather forecasting.

Starting with the first, we have noticed that experimental results shown significant correlation of the system performance and the initialization of several system parameters. It is then necessary to investigate methods for optimized initialization of the training process. A good approach in this direction would be the use of other features, such as formants or other articulatory features for the initialization of the state space. Along with the different types of features, it would be interesting to examine different dimensions of the state vector and/or observation vector, too.

In implementation of the LDSSMs for speech recognition, we used frame-segment alignments for training the word-models. We have chosen to simplify our implementation at that phase and obtain these alignments using HTK and equivalent, well trained HMMs consisting of as many states as the number of segments defined in each of the word-models. We believe that a more sophisticated approach to this would be to develop algorithms to obtain feature-segments alignments through dynamic programming.

In theory, we investigated the use of an extra control input for the proposed GISSMs. It should be challenging for our research to examine how this could be useful in speech recognition. Could it be possible to use an extra input in a speech recognition task? What would that be? A clear answer at the moment, could not be said. Maybe we could use the air flow between the mouth of the speaker and the

microphone, as an extra input.

Finally, we applied our model only on the clean set of AURORA 2 database. However, the real challenge would be to evaluate the performance of LDSSM on the rest of the database, with different noise additions in various noisy scales. The power of segment-based models is in these kind of situations.

Results in weather prediction shown also that there are some subjects that could be improved. We only used 360 samples for training and 180 for testing. Maybe it would be better to obtain from a meteorological service, data for more than three years in order to more efficiently train the month-models and to evaluate the prediction performance of the system with larger data set.

In our experiments on weather forecasting, we only used temperature samples obtained in the morning, midday and night. We can investigate the use of extra information by rising the observation vector and the state vector, too. For a more accurate weather prognoses, one would need information about the humidity, the wind, pressure, dewpoint and many other meteorological parameters that meteorologists take into consideration when they train their models to predict the weather.

In conclusion, we have applied the GISSM only on the first half of the year, months from January to June. One of the reasons for doing this was the "temperature similarity" between Spring and Autumn, and also among July and August, in Chania. Taking under consideration additional parameters, as those mentioned above, we could apply the model to the whole year since it could be possible to obtain more robust models for each month and hence to be able to predict weather with a better performance. In all cases, we cannot ignore weather's non-linearity and that is a fact that always would cause error in prognoses.

Appendix A

Derivation of the Element-wise estimation of General State-Space models' parameters

A.1 Without control input

In this appendix we derive the Maximum-Likelihood estimates in an element-wise way of the parameter matrix F and the state covariance P in case with no extra control input and in next section, in case of additional input B.

Proposition A.1.1. *If matrix F has the identifiable canonical form shown in 3.2.1 and the covariance P is a full matrix of the same dimension as F, then the element-wise estimators of F, P that maximize the quantity*

$$L(\mathbf{X}, \mathbf{Y}) = -\frac{N}{2}\log|P| - \frac{1}{2}\sum_{k=1}^{N}\{(y_k - Fx_k)^T P^{-1}(y_k - Fx_k)\} + constant \quad \text{(A.1.1)}$$

are given by

$$\hat{F}^{i,j} = \frac{\sum_{c=1}^{M}\left\{(cof(\hat{P}^{i,c}))(\Gamma_4^{c,j})\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1,c\neq i}^{M}\left\{(cof(\hat{P}^{i,c}))(\hat{F}^{c,j})(\Gamma_3^{j,j})\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\left\{(cof(\hat{P}^{i,c}))\sum_{r=1,r\neq j}^{M}\left\{(\hat{F}^{c,r})(\Gamma_3^{r,j})\right\}\right\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} \quad \text{(A.1.2)}$$

$$\hat{P}^{i,j} = (\Gamma_2^{i,j}) - \sum_{r=1}^{M}(\hat{F}^{i,r})(\Gamma_4^{j,r}) - \sum_{r=1}^{M}(\hat{F}^{j,r})(\Gamma_4^{i,r}) + \sum_{c=1}^{M}\sum_{r=1}^{M}(\hat{F}^{i,c})(\hat{F}^{j,r})(\Gamma_3^{c,r}) \quad \text{(A.1.3)}$$

Proof. The likelihood function that we want to maximize is

$$L(\mathbf{X},\mathbf{Y}) = -\frac{N}{2}\log|P| - \frac{1}{2}\sum_{k=1}^{N}\{(y_k - Fx_k)^T P^{-1}(y_k - Fx_k)\} + \text{constant}. \quad \text{(A.1.4)}$$

In Digalakis PhD thesis [12] has been derived that the quantity (A.1.4) can be maximized by

$$\hat{\theta} = \mathtt{a}rgmin \det\left[\frac{1}{N}\sum_{k=1}^{N} e_k(\theta)e_k(\theta)^T\right] \quad \text{(A.1.5)}$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N} e_k(\theta)e_k(\theta)^T. \quad \text{(A.1.6)}$$

Hence, we can write:

$$\frac{\partial\hat{F}}{\partial F^{i,j}} = \frac{\partial|P|}{\partial F^{i,j}} = \sum_{r=1}^{M}\sum_{c=1}^{M}\frac{\partial|P|}{\partial P^{r,c}}\frac{\partial P^{r,c}}{\partial F^{i,j}}. \quad \text{(A.1.7)}$$

We know that

$$|P| = \sum_{c=1}^{M} P^{r,c}\, cof(P^{r,c}) \quad \text{(A.1.8)}$$

but $cof(P^{r,c})$ is independent of $P^{r,c}$ for every (r,c), and so

$$\frac{\partial |P|}{\partial P^{r,c}} = \frac{\partial \sum_{c=1}^{M} P^{r,c}\, cof(P^{r,c})}{\partial P^{r,c}} = cof(P^{r,c}). \tag{A.1.9}$$

Hence, finally we get

$$\frac{\partial |P|}{\partial F^{i,j}} = \sum_{r=1}^{M}\sum_{c=1}^{M} cof(P^{r,c}) \frac{\partial P^{r,c}}{\partial F^{i,j}}. \tag{A.1.10}$$

It can be shown, using some elementary matrix algebra, that the derivative $\frac{\partial P}{\partial F^{i,j}}$ is given by

$$\frac{\partial P}{\partial F^{i,j}} = 2\sum_{r=1}^{M} F^{i,r} x^r x^j - 2y^i x^j \tag{A.1.11}$$

and considering the (A.1.10) we get for the general case the following equation

$$\frac{\partial |P|}{\partial F^{i,j}} = 2\sum_{c=1}^{M}\left\{ cof(P^{i,c}) \left[\sum_{r=1}^{M} (F^{c,r} x^r x^j) - y^c x^j \right] \right\}. \tag{A.1.12}$$

If we consider the above to be equal to zero we finally get the equation (A.1.2).

For the covariance matrix we have

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N} e_k(\theta)e_k(\theta)^T \iff$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{(y_k - Fx_k)(y_k - Fx_k)^T\Big\} \iff$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{\Big(y_k - \begin{bmatrix} F^i & x_k \end{bmatrix}\Big)\Big(y_k - \begin{bmatrix} F^i & x_k \end{bmatrix}\Big)^T\Big\} \iff$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{y_k y_k^T - \begin{bmatrix} F^i & x_k \end{bmatrix} y_k^T - y_k \begin{bmatrix} F^i & x_k \end{bmatrix}^T + \begin{bmatrix} F^i & x_k & x_k^T & (F^i)^T \end{bmatrix}\Big\} \iff$$

$$\hat{P}^{i,j} = \frac{1}{N}(\sum_{k=1}^{N} y_k^i y_k^j) - \frac{1}{N}\sum_{r=1}^{M}\Big\{(\hat{F}^{i,r})(\sum_{k=1}^{N} x_k^r y_k^j)\Big\} - \frac{1}{N}\sum_{r=1}^{M}\Big\{(\hat{F}^{j,r})(\sum_{k=1}^{N} x_k^r y_k^i)\Big\} +$$
$$\frac{1}{N}\sum_{c=1}^{M}\sum_{r=1}^{M}\Big\{(\hat{F}^{i,c})(\hat{F}^{j,r})(\sum_{k=1}^{N} x_k^c x_k^r)\Big\}. \tag{A.1.13}$$

□

A.2 With control input

In this section we derive the Maximum-Likelihood estimates in an element-wise way when we have control input.

Proposition A.2.1. *If matrix F has the identifiable canonical form shown in (3.2.1) and the control matrix B the one that shown in (3.2.5) and the covariance P are filled of free parameters, then the element-wise estimators of F, B, and P that maximize*

the quantity

$$J(\mathbf{X},\mathbf{Y},\theta) = -L(\mathbf{X},\mathbf{Y},\theta) = \sum_{k=1}^{N}\Big\{\log|P| + (x_k - Fx_{k-1} - Bu_{k-1})^T P^{-1}(x_k - Fx_{k-1} - Bu_{k-1})\Big\} + \sum_{k=0}^{N}\Big\{\log|R| + (y_k - Hx_k)^T R^{-1}(y_k - Hx_k)\Big\} + constant \tag{A.2.1}$$

are given by

$$\hat{F}^{i,j} = \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))(\Gamma_4^{c,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{q=1}^{T}\Big\{(\hat{B}^{c,q})(\Gamma_8^{q,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1,c\neq i}^{M}\Big\{(cof(\hat{P}^{i,c}))(\hat{F}^{c,j})(\Gamma_3^{j,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{r=1,r\neq j}^{M}\Big\{(\hat{F}^{c,r})(\Gamma_3^{r,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_3^{j,j})} \tag{A.2.2}$$

$$\hat{B}^{i,j} = \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))(\Gamma_{10}^{c,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{s=1}^{M}\Big\{(\hat{F}^{c,s})(\Gamma_7^{s,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \frac{\sum_{c=1,c\neq i}^{M}\Big\{(cof(\hat{P}^{i,c}))(\hat{B}^{c,j})(\Gamma_9^{j,j})\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} - \frac{\sum_{c=1}^{M}\Big\{(cof(\hat{P}^{i,c}))\sum_{q=1,q\neq j}^{T}\Big\{(\hat{B}^{c,q})(\Gamma_9^{q,j})\Big\}\Big\}}{(cof(\hat{P}^{i,i}))(\Gamma_9^{j,j})} \tag{A.2.3}$$

$$\hat{P}^{i,j} = (\Gamma_2^{i,j}) - \sum_{r=1}^{M}(\hat{F}^{i,r})(\Gamma_4^{j,r}) - \sum_{r=1}^{M}(\hat{F}^{j,r})(\Gamma_4^{i,r}) + \sum_{c=1}^{M}\sum_{r=1}^{M}(\hat{F}^{i,c})(\hat{F}^{j,r})(\Gamma_3^{c,r}) - \sum_{q=1}^{T}(\hat{B}^{i,q})(\Gamma_{10}^{j,q}) + \sum_{q=1}^{T}\sum_{r=1}^{M}(\hat{B}^{i,q})(\hat{F}^{j,r})(\Gamma_8^{q,r}) - \sum_{q=1}^{T}(\hat{B}^{j,q})(\Gamma_{10}^{i,q}) + \sum_{r=1}^{M}\sum_{q=1}^{T}(\hat{F}^{i,r})(\hat{B}^{j,q})(\Gamma_7^{r,q}) + \sum_{q=1}^{T}\sum_{p=1}^{T}(\hat{B}^{i,q})(\hat{B}^{j,p})(\Gamma_9^{q,p}) \tag{A.2.4}$$

Proof. The likelihood function that we want to maximize is given by (A.2.1). Following the same steps as in the previous section, we again conclude in equation (A.1.10). At this time, the derivative of $|P|$ in respect to the $ij-th$ element of F is given by the following

$$\frac{\partial |P|}{\partial F^{i,j}} = 2\sum_{c=1}^{M}\Bigg\{cof(P^{i,c})\Bigg[\sum_{r=1}^{M}(F^{c,r}x^r x^j) + \sum_{q=1}^{T}(B^{c,q}u^q x^j) - y^c x^j\Bigg]\Bigg\}. \tag{A.2.5}$$

If we consider the above to be equal to zero we finally get the equation (A.2.2).

In the same way, the equation that we get if we consider derivation with respect to $B^{i,j}$, is given by

$$\frac{\partial |P|}{\partial B^{i,j}} = 2\sum_{c=1}^{M}\Bigg\{cof(P^{i,c})\Bigg[\sum_{r=1}^{M}(F^{c,r}x^r u^j) + \sum_{q=1}^{T}(B^{c,q}u^q u^j) - y^c u^j\Bigg]\Bigg\}. \tag{A.2.6}$$

By considering again, the above equation to be equal to zero, we conclude in (A.2.3).

For the covariance matrix we have

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N} e_k(\theta)e_k(\theta)^T \Longleftrightarrow$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{(y_k - Fx_k - Bu_k)(y_k - Fx_k - Bu_k)^T\Big\} \Longleftrightarrow$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{\Big(y_k - \begin{bmatrix} F^i & x_k \end{bmatrix} - \begin{bmatrix} B^i & u_k \end{bmatrix}\Big)\Big(y_k - \begin{bmatrix} F^i & x_k \end{bmatrix} - \begin{bmatrix} B^i & u_k \end{bmatrix}\Big)^T\Big\} \Longleftrightarrow$$

$$\hat{P} = \frac{1}{N}\sum_{k=1}^{N}\Big\{y_k y_k^T - \begin{bmatrix} F^i & x_k \end{bmatrix} y_k^T - \begin{bmatrix} B^i & u_k \end{bmatrix} y_k^T$$

$$-y_k \begin{bmatrix} F^i & x_k \end{bmatrix}^T + \begin{bmatrix} F^i & x_k & x_k^T & (F^i)^T \end{bmatrix} + \begin{bmatrix} B^i & u_k & x_k^T & (F^i)^T \end{bmatrix}$$

$$-y_k \begin{bmatrix} B^i & u_k \end{bmatrix}^T + \begin{bmatrix} F^i & x_k & u_k^T & (B^i)^T \end{bmatrix} + \begin{bmatrix} B^i & u_k & u_k^T & (B^i)^T \end{bmatrix}\Big\} \Longleftrightarrow$$

$$\hat{P}^{i,j} = \frac{1}{N}(\sum_{k=1}^{N} y_k^i y_k^j) - \frac{1}{N}\sum_{r=1}^{M}\Big\{(\hat{F}^{i,r})(\sum_{k=1}^{N} x_k^r y_k^j)\Big\} - \frac{1}{N}\sum_{r=1}^{M}\Big\{(\hat{F}^{j,r})(\sum_{k=1}^{N} x_k^r y_k^i)\Big\}$$

$$+\frac{1}{N}\sum_{c=1}^{M}\sum_{r=1}^{M}\Big\{(\hat{F}^{i,c})(\hat{F}^{j,r})(\sum_{k=1}^{N} x_k^c x_k^r)\Big\} - \frac{1}{N}\sum_{q=1}^{T}\Big\{(\hat{B}^{i,q})(\sum_{k=1}^{N} u_k^q y_k^j)\Big\}$$

$$+\frac{1}{N}\sum_{q=1}^{T}\sum_{r=1}^{M}\Big\{(\hat{B}^{i,q})(\hat{F}^{j,r})(\sum_{k=1}^{N} u_k^q x_k^r)\Big\} - \frac{1}{N}\sum_{q=1}^{T}\Big\{(\hat{B}^{j,q})(\sum_{k=1}^{N} u_k^q y_k^i)\Big\}$$

$$+\frac{1}{N}\sum_{r=1}^{M}\sum_{q=1}^{T}\Big\{(\hat{F}^{i,r})(\hat{B}^{j,q})(\sum_{k=1}^{N} x_k^r u_k^q)\Big\} + +\frac{1}{N}\sum_{q=1}^{T}\sum_{p=1}^{T}\Big\{(\hat{B}^{i,q})(\hat{B}^{j,p})(\sum_{k=1}^{N} u_k^q u_k^p)\Big\}. \quad \text{(A.2.7)}$$

□

Bibliography

[1] C. R. Athaide. *Likelihood Evaluation and State Estimation for Nonlinear State Space Models*. PhD thesis, Graduate Group in Managerial Science and Applied Economics, University of Pennsylvania, Philadelphia, PA, USA, 1995.

[2] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *Readings in speech recognition*, pages 308–319, 1990.

[3] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[4] E. Bocchieri and G. Doddington. Frame-specific statistical features for speaker-independent speech recognition. *IEEE Trans. Acoust. Speech and Signal Processing*, 34(4):755–764, August 1986.

[5] E. L. Bocchieri. Vector quantization for the efficient computation of continuous density likelihoods. *IEEE Proc. of Internat. Conf. on Acoustics, Speech, and Signal Processing, Minneapolis, Minnesota, USA*, pages 692–694, April 1993.

[6] H. Bourlard. A new training algorithm for statistical sequence recognition with applications to transition-based speech recognition. *IEEE Signal Processing Letters*, 3:203–205, 1996.

[7] M. A. Bush and G. E. Kopec. Network-based connected digit recognition. *IEEE Trans. Acoust. Speech and Signal Processing*, 35(10):1401–1413, October 1987.

[8] P. E. Caines. *Linear Stochastic Systems*. John Wiley & Sons, 1998.

[9] G. Cook and A. Robinson. Transcribing broadcast news with the 1997 abbot system. *Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, WA, USA*, pages 917–920, 1998.

[10] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistical Society (B)*, 39(1):1–38, 1977.

[11] L. Deng and J. Ma. A statistical coarticulatory model for the hidden vocal-tract-resonance dynamics. *In Proceedings Eurospeech*, 4:1499–1502, 1999.

[12] V. Digalakis. *Segment-based stochastic models of spectral dynamics for continuous speech recognition*. PhD thesis, Boston University, January 1992.

[13] V. Digalakis, J. R. Rohlicek, and M. Ostendorf. Ml estimation of a stochastic linear system with the em algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(4), October 1993.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[15] K. Erler and G. H. Freeman. An HMM-based speech recognizer using overlapping articulatory features. *J. Acoust. Soc. Amer.*, 100:2500–2513, 1996.

[16] A. J. Robinson et al. A neural network based, speaker independent, large vocabulary. *Proc. of the European Conf. on Speech Communication and Technology, Berlin, Germany*, pages 1941–1944, 1999.

[17] G. Zavaliagkos et al. A hybrid segmental neural net/hidden markov model system for continuous speech recognition. *IEEE Trans. on Speech and Audio Processing*, 2:151–160, 1994.

[18] G. Zue et al. Recent progress on the summit system. *Proceedings of the Third DARPA Workshop on Speech and Natural Language*, June 1990.

[19] R. Cole et.al. *Survey of the state of the Art in Human Language Technology (Studies in Natural Language Processing)*. Cambridge University Press, March 1998.

[20] J. Frankel. *Linear dynamic models for automatic speech recognition*. PhD thesis, The Centre for Speech Technology Research, Edinburgh University, 2003.

[21] J. Fritsch and M. Finke. ACID/HNN: Clustering hierarchies of neural networks for context-dependent connectionist acoustic modeling. *Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, WA, USA*, pages 505–508, 1998.

[22] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. *University of Toronto Technical Report CRG-TR-96-2*, 1996.

[23] Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):831–864, 2000.

[24] N. K. Gupta and R. K. Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Trans. Automatic Control*, AC-19(6):774–783, 1974.

[25] J. Hennebert. Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. *Proc. of the Eurospeech Conf., Rhodes, Greece*, pages 1951–1954, 1997.

[26] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.

[27] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001.

[28] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556, April 1976.

[29] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, Series D, J. Basic Eng.*, 82:35–45, March 1960.

[30] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, Series D, J. Basic Eng.*, 83:95–108, 1961.

[31] R. L. Kashyap. Maximum likelihood identification of stochastic linear systems. *IEEE Trans. Automatic Control*, AC-15(1):25–34, February 1970.

[32] C.-J. Kim. Dynamic linear models with markov-switching. *Journal of Econometrics*, 60:1–22, 1994.

[33] O. A. Kimball. *Segment Modeling Alternatives for Continuous Speech Recognition*. PhD thesis, Boston University, 1995.

[34] L.J. Lee, H. Attias, and L. Deng. Variational inference and learning for segmental switching state space models of hidden speech dynamics. *In Proceedings ICASSP*, 1:920–923, 2003.

[35] L. Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, December 1998.

[36] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, Massachussets, USA, 1983.

[37] L. Ljung and G. Torkel. *Modeling of Dynamic Systems*. Prentice Hall PTR, Information and system sciences series, 1994.

[38] David G. Luenberger. *Introduction to Dynamic Systems. Theory, models and Applications*. John Wiley & Sons, Inc, 1979.

[39] J. Ma and L. Deng. Optimization of dynamic regimes in a statistical hidden dynamic model for conversational speech recognition. *In Proceedings Eurospeech*, 3:1339–1342, 1999.

[40] J. Ma and L. Deng. A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech. *Computer Speech and Language*, 14(2):101–114, 2000.

[41] J. Ma and L. Deng. Efficient decoding strategy for conversational speech recognition using state-space models for vocal-tract-resonance dynamics. *In Proceedings Eurospeech*, pages 603–606, 2001.

[42] J. Ma and L. Deng. A mixed-level switching dynamic system for continuous speech recognition. *Computer Speech and Language*, 18(1):49–65, 2004.

[43] S. Makino and K. Kido. Recognition of phonemes using time-spectrum pattern. *Speech Communication*, 5:115–133, 1943.

[44] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(2):225–238, June 1986.

[45] N. Merhav and Y. Ephraim. Hidden markov modeling using the most likely state sequence. *IEEE International Conf. Acoust. Speech Signal Processing, Toronto, Canada*, pages 469–472, May 1991.

[46] N. Morgan and H. Bourlard. Continuous speech recognition: An introduction to hybrid HMM/Connectionist approach. *IEEE Signal Processing Magazine*, pages 25–42, 1995.

[47] M. Ostendorf and V. Digalakis. The stochastic segment model for continuous speech recognition. *In Proceedings The 25th Asilomar Conference on Signals, Systems and Computers*, pages 964–968, 1991.

[48] M. Ostendorf, V. Digalakis, and O.A. Kimball. From hmms to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, 1996.

[49] M. Ostendorf, A. Kannan, O. Kimball, and J. R. Rohlicek. Continuous word recognition based on the stochastic segment model. *in Proc. DAPRA Workshop on CSR*, 1992.

[50] M. Ostendorf and S. Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Trans. Acoustic Speech and Signal Processing*, 37(12):1857–1869, December 1989.

[51] J. Picone, S. Pike, R. Regan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, and M. Schuster. Initial evaluation of hidden dynamic models on conversational speech. *In Proceedings ICASSP*, 1:109–112, 1999.

[52] L. A. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[53] H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372, 1963.

[54] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

[55] H. B. Richards and J. S. Bridle. The hdm: A segmental hidden dynamic model of coarticulation. *In Proceedings ICASSP*, 1:357–360, 1999.

[56] M. Richardson, J. Bilmes, and C. Diorio. Hidden-articulator markov models for speech recognition. *ASR*, pages 133–139, 2000.

[57] M. Richardson, J. Bilmes, and C. Diorio. Hidden-articulator markov models: performance improvements and robustness to noise. *ICSLP*, 3:131–134, 2000.

[58] M. Richardson, J. Bilmes, and C. Diorio. Hidden-articulator markov models for speech recognition. *Speech Communication*, 41:511–529, 2003.

[59] A. J. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Trans. on Neural Networks*, 5:298–305, 1994.

[60] Antti-Veikko Ilmari Rosti. *Linear Gaussian Models for Speech Recognition*. PhD thesis, University of Cambridge, Wolfson College, May 2004.

[61] S. Roucos, M. Ostendorf, H. Gish, and A. Derr. Stochastic segment modeling using the estimate-maximize algorithm. *IEEE Int. Conf. Acoust. Speech Signal Processing, New York, NY, USA*, pages 127–130, April 1988.

[62] S. Roweis and Z. Ghahramani. A unified review of the linear gaussian models. *Neural Computation*, 11(2), 1999.

[63] D. E. Rummelhart and J. L. McClelland. *Parallel Distributed Processing - Explorations in the Microstructure of Cognition, Volume I : Foundations*. Cambridge, MA, MIT Press., 1986.

[64] S. Russel and P. Norving. *Artificial Intelligence: A modern Approach*. Prentice Hall PTR, Englewood Cliffs, NJ, USA, 1995.

[65] F. Seide, J. L. Zhou, and L. Deng. Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM - MAP decoding and evaluation. *In Proceedings ICASSP*, 1:748–751, 2003.

[66] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[67] R. H. Shumway and D. S. Stoffer. Dynamic linear models with switching. *Journal of American Stat. Assoc.*, 86:763–769, 1991.

[68] A. H. Waibel and K. F. Lee. *Readings in Speech Recognition*. Morgan Kaufman Publishers, San Mateo, CA, USA, 1990.

[69] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.2)*. Cambridge University, Engineering Department, December 2002.

[70] J. L. Zhou, F. Seide, and L. Deng. Coarticulation modeling by embedding a target-directed hidden trajectory model into hmm - model and training. *In Proceedings ICASSP*, 1:744–747, 2003.