A State Space World for Detecting and Estimating Events and Learning Sparse Signal Decompositions

Doctoral Thesis

Author(s):
Zalmai, Nour

Publication date:
2017

Permanent link:
https://doi.org/10.3929/ethz-b-000176652

Rights / license:
In Copyright - Non-Commercial Use Permitted
A State Space World for Detecting and Estimating Events and Learning Sparse Signal Decompositions

A dissertation submitted to
ETH Zurich
for the degree of
Doctor of Sciences

presented by
Nour Zalmaï
M.Sc. ETH Zurich, Dipl. Ing. Supélec
born on September 12, 1989
citizen of France

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Bhaskar D. Rao, co-examiner

2017
To my Family and my Friends

“Je prends beaucoup plus de plaisir à m’instruire moi-même que non pas à mettre par écrit le peu que je sais.”

“I take much more pleasure in educating myself rather than writing down what little I know.”

René Descartes
Acknowledgments

I could write a whole chapter of acknowledgements but my clairvoyance dictates me to be concise and precise.

First of all, I would like to thank all the people who directly contributed to this thesis. My work was highly influenced by my master thesis supervisors Christoph Reller and Lukas Bruderer. I am also deeply indebted to Reto Wildhaber and Hampus Malmberg for very fruitful collaborations. The current exposition of this thesis would not have been possible without their constant feedback and complaints. I must also recognize that I was fortuitous to supervise seven bright students in their master thesis: Christian Käslin, Désirée Clausen, Andreas Büchel, Clément Luneau, Raphael Keusch, Timon Künzle, and Elizabeth Ren. All of them were motivated, fearless, and hard workers. I still do not know whether I taught them more than I learned from them: they were often questioning theory, which made me think, explore, and discover new facets and prospects. What is certain is that the variety of applications included in this thesis was reachable through their hard work. I am also grateful to Bhaskar Rao for taking the time to read and examine my thesis and inspiring me with sparse Bayesian learning. Last but not least, this thesis is a reflection of the constant support from my thesis supervisor Hans-Andrea Loeliger. I want to sincerely thank Andi for the freedom he gave me during my PhD studies; this freedom was necessary to find my own way into the state space world. I will also always remember his endless enthusiasm and conviction to prove any formula with an elegant graphical proof that avoids the use of sums and indices.

Secondly, I would like to thank all the people in my working environment who, to a certain extent, also contributed to this thesis. I want to thank all my colleagues and in particular, Sarah Neff, for several interesting discussions, Christian Schürch, for his objectivity and honesty in any circumstances, Annina Bracher, for her analytical skills and her
vision of the world, David Stotz, for providing me tips for solving my mathematical problems, Céline Aubel, for numerous and passionate discussions, Hampus Malmberg, for pseudo-randomly mixing concepts to generate new awesome ideas, Reto Wildhaber, for his extra-ordinary intuition that any theoretical scientist should envy, Michael Tschannen, for being so special, Federico Wadehn, for bringing up interesting biomedical problems and kicking Reto into my office, Patrick Murer, for enlightening us with his presence every single day, Tibor Keresztfalvi, for being always there when we need him, Christoph Pfister, for his passion about how things work, Boxiao Ma, for smiling all the time and taking the time to listen, and Anna Volokitin, for sharing enjoyable coffee breaks and reminding me of simple things I tend to forget. I am also grateful to the fundamental pillars of the ISI institute and in particular, Jiun-Hung Yu, for his endless motivation for research, Stefan Moser, for caring about things by expressing meaningful complaints, Amos Lapidoth, for his sense of humor and his constant cheering, and Paddy and Rita, for helping you whenever you need them. I also want to thank all the students I supervised (the list is too long to be enumerated) and especially Gabrielle Buytaert. With all these people, we enjoyed several activities that have become traditions at ISI: baby-foot, AoFE, soccer game, badminton, pizzas, running at SOLA, and many others. Furthermore, I cannot resist from acknowledging the IT support group at D-ITET for their amazing maintenance work for computers, systems, and printers. This seems to be the necessary evil that teaches you how to manage things yourself.

Third, I would like to thank all the people I have met during my time in Zurich. Special thanks to my friends from my master studies at ETH and in particular to Bertrand Pouymayou. When playing badminton in order to change my mind and temporarily leave the state space world, I met very good friends and in particular, Julia, Yael, Esther, Rea, Yiran, and Jasmina. All of them are very funny and shared their joie de vivre. I also enjoyed swimming with my entertaining friend Daniela.

Finally, I would like to sincerely thank all my family and especially my parents, Julienne and Ghafori, and my siblings, Loria, David, and Nazir. They have always trusted me, believed in me, and supported me in all my choices even if they could not understand what I was actually doing and what I would end up doing. This is in contrast to the time when I was a kid and when I was always told to naively work hard if I wanted to make my dreams come true. Lastly, I am grateful to the French education system which provided me with a high-quality education.
Abstract

Many signals can be labeled with a small set of events such that each event is categorized according to its surrounding signal shapes. In this thesis, we provide a general approach based on linear state space models to learn sparse signal decompositions from single-channel and multi-channel discrete-time measurements. The proposed approach provides a sparse multi-channel representation of a given signal, which can be interpreted as a signal labeling. This thesis is organized in three parts.

In the first part, several important properties of linear state space models (LSSMs) are revisited. Especially, signals generated with an autonomous LSSM are thoroughly investigated and fully characterized. In particular, we show that the set of such signals forms a ring and that the correlation function between any autonomous LSSM signal and any discrete-time signal can be recursively and efficiently computed. These two properties along with the vast modeling capabilities of LSSM signals are at the heart of this thesis.

In the second part, we develop a general approach to detect events in (single-channel or multi-channel) discrete-time signals and estimate the parameters of such events. Since the number of events is assumed to be substantially smaller than the number of samples, the set of detected events is interpreted as a sparse representation of the given signal. An event locally creates characteristic signals which are modeled with a two-sided autonomous LSSM; the right-sided model accounts for the signals observed after that event while the left-sided model accounts for the signals before that event. Thus, the problem of event detection and estimation is substituted by fitting at any given time a LSSM signal to observations. For this purpose, new cost functions are defined: a LSSM-weighted squared error cost and a LSSM-weighted polynomial cost. These cost functions have the attractive property of being recursively computed. In addition, closed-form solutions for several mini-
mization problems are available. As far as event detection is concerned, several hypothesis tests with a suitable notion of local likelihood are promoted. Surprisingly, event detection in various conditions, such as in the presence of an unknown additive or multiplicative interference signal, can be naturally dealt with. Finally, various important practical applications are addressed in detail in order to exemplify the potential of the proposed approach for event detection and estimation.

In the third and last part, we propose a general approach to learn sparse signal decompositions. We assume that each signal component can be sparsely represented in the input domain of some unknown LSSM. We model sparse inputs with zero-mean Gaussian random variables with unknown variances, as in the sparse Bayesian learning framework. Then, all unknown parameters are estimated by maximum likelihood with an expectation maximization (EM) algorithm where all parameters are jointly updated with closed-form expressions and all expectation quantities are efficiently computed with a Gaussian message passing algorithm. This general approach can deal with a large variety of sparse signal decomposition problems. Among them, we address the problems of learning repetitive signal shapes, learning classes of signal shapes, and decomposing a signal with scaled, time-shifted, and time-dilated versions of a signal shape. All these concepts and methods are illustrated with practical examples.

**Keywords:** Linear state space models; event detection and estimation; learning sparse signal decompositions; sparse Bayesian learning; unsupervised feature extraction.
Kurzfassung


Im zweiten Teil entwickeln wir einen allgemeinen Ansatz, um in (ein- oder mehrkanaligen) zeitdiskreten Signalen Ereignisse zu detektieren und ihre dazugehörenden Parameter zu schätzen. Die Annahme ist, dass die Anzahl Ereignisse deutlich kleiner ist, als die Anzahl der Samples. Dadurch wird die Menge der erkannten Ereignisse als eine sparse Repräsentation des ursprünglichen Signals aufgesfasst. Jedes Ereignis erzeugt lokal charakteristische Signale, welche durch ein zweiseitiges autonomes LSSM modelliert werden. Dabei repräsentiert das rechtsseitige Modell die Signale nach dem Ereignis und das linksseitige Modell die Signale vor dem Ereignis. Damit werden die Ereignisse durch ein zu jedem Zeitindex durchgeführtes LSSM-fitting (d.h. anpassen eines LSSMs


**Stichworte:** Lineare Zustandsraummodelle; Ereignis Detektion und Schätzung; learning sparse signal decompositions; sparse Bayesian learning; Unüberwachtes Extrahieren von Merkmalen.
Résumé

De nombreux signaux peuvent être annotés par un ensemble clairsemé d’événements où chaque événement est catégorisé selon les formes de ses signaux environnants. Dans cette thèse, nous développons une approche générale qui repose sur les modèles linéaires à représentation d’état, pour l’apprentissage de décompositions parcimonieuses de signaux à partir d’observations à temps discret à un ou plusieurs canaux. L’approche proposée fournit une représentation parcimonieuse à plusieurs canaux d’un signal donné. Cette représentation peut être interprétée comme une annotation du signal. Cette thèse s’articule autour de trois parties.

Dans la première partie, nous rappelons plusieurs propriétés importantes des modèles linéaires à représentation d’état (abrégés, LSSMs pour « linear state space models »). Plus particulièrement, nous étudions de manière extensive les signaux générés par un LSSM autonome et nous les caractérisons précisément. En particulier, nous montrons que l’ensemble de ces signaux a une structure d’anneau et que nous pouvons calculer récursivement et de manière efficace la fonction de corrélation entre un signal LSSM autonome et tout signal à temps discret. Ces deux propriétés, auxquelles s’ajoute la grande aptitude de modélisation des signaux générés par un LSSM, sont au cœur de cette thèse.

Dans la deuxième partie, nous développons une approche générale pour détecter des événements et estimer leurs paramètres à partir d’un signal à temps discret à un ou plusieurs canaux. Puisque le nombre d’événements est supposé être bien plus faible que le nombre d’échantillons, l’ensemble des événements détectés peut être interprété comme une représentation parcimonieuse du signal en question. Un événement crée localement des signaux caractéristiques que nous modélisons avec un LSSM autonome bidirectionnel; le modèle se propageant vers la droite représente les signaux observés après l’événement tandis que le modèle se propageant vers la gauche représente les signaux avant l’événement.
Ainsi, nous substituons le problème de détection et d’estimation d’événements par l’ajustement, à tout instant, d’un LSSM aux observations. Dans ce but, nous introduisons de nouvelles fonctions de coût : un coût quadratique pondéré par un signal LSSM et un coût polynomial pondéré par un signal LSSM. Ces fonctions de coût ont la propriété attrayante de pouvoir être calculées récursivement. De plus, des solutions explicites pour plusieurs problèmes de minimisation sont disponibles. En ce qui concerne la détection d’événements, plusieurs tests d’hypothèses, associés à une notion adéquate de vraisemblance locale, sont présentés. Étonnamment, nous pouvons détecter soigneusement des événements dans des conditions variées, telles que la présence d’un signal inconnu interférant additivement ou multiplicativement. Enfin, de nombreuses applications pratiques sont présentées dans le but de mettre en lumière le potentiel de l’approche que nous proposons pour la détection et l’estimation d’événements.

Dans la troisième et dernière partie, nous proposons une approche générale pour l’apprentissage de décompositions parcimonieuses de signaux. Nous supposons que chaque composante du signal peut être représentée de manière parcimonieuse dans le domaine d’entrée d’un LSSM inconnu. Nous modélisons le signal d’entrée parcimonieux avec des variables aléatoires Gaussiennes à moyenne nulle avec des variances inconnues, comme dans la méthode d’apprentissage Bayésien parcimonieux (« sparse Bayesian learning »). Ensuite, tous les paramètres inconnus sont estimés en maximisant la vraisemblance avec un algorithme EM (« expectation maximization ») où tous les paramètres sont conjointement actualisés avec des formules explicites et où toutes les espérances des variables aléatoires sont efficacement calculées grâce à un algorithme de passage de messages Gaussiens (« Gaussian message passing »). Cette approche générale peut résoudre une grande variété de problèmes de décomposition parcimonieuse de signaux. Parmi eux, nous abordons les problèmes d’apprentissages de formes répétées, d’apprentissage de classes de formes, et de décomposition d’un signal avec une forme et ses versions mises à l’échelle, décalées dans le temps et dilatées. Tous ces concepts et ces méthodes sont illustrés par des exemples pratiques.

Mots clefs : Systèmes linéaires à représentation d’état; détection et estimation d’événements; apprentissage de décompositions parcimonieuses de signaux; apprentissage Bayésien parcimonieux; extraction non-supervisée d’informations caractéristiques.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>vii</td>
</tr>
<tr>
<td>Kurzfassung</td>
<td>ix</td>
</tr>
<tr>
<td>Résumé</td>
<td>xi</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>xix</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Outline and Contribution</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Related Work</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Acknowledgments</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Notation</td>
<td>8</td>
</tr>
<tr>
<td><strong>I Linear State Space Models</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>2 Generalities on Linear State Space Models</strong></td>
<td>11</td>
</tr>
<tr>
<td>2.1 Solving the LSSM Equations</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Statistical Assumptions</td>
<td>12</td>
</tr>
<tr>
<td>2.3 LSSM Inputs Fed to a LSSM</td>
<td>15</td>
</tr>
<tr>
<td>2.4 LSSM Forward and Backward in Time</td>
<td>16</td>
</tr>
<tr>
<td>2.4.1 LSSM Backward in Time</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2 LSSM Forward/Backward in Time</td>
<td>16</td>
</tr>
<tr>
<td><strong>3 Autonomous Linear State Space Models</strong></td>
<td>17</td>
</tr>
<tr>
<td>3.1 Definitions and Outline</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Algebraic Structure of the Set of LSSM Signals</td>
<td>18</td>
</tr>
<tr>
<td>3.3 Equivalent LSSM Signal Form</td>
<td>19</td>
</tr>
</tbody>
</table>
### Contents

3.3.1 From a Signal Form to a LSSM Parameterization . 20  
3.3.2 From a LSSM Parametrization to a Signal Form . 24  
3.4 Uniqueness of LSSM Signal Form . 25  
3.5 Initial State Versus Observation Vector . 27  
3.6 A Change of Time-Scale . 29  
3.7 Fitting a LSSM Signal . 30  
3.8 Inner Product With LSSM Signals . 34  

4 Computations in Linear State Space Models . 37  
4.1 Gaussian Message Passing . 38  
4.1.1 Generality . 38  
4.1.2 Modified Bryson-Frazier Smoother . 39  
4.1.3 Improving Numerical Stability . 41  
4.1.4 Processing LSSM Models defined Forward and Backward in Time . 43  
4.2 From Messages to Expectations . 44  
4.3 Likelihood Computation . 46  

II Detection and Estimation in Multi-Channel Signals With Autonomous Linear State Space Models . 47  

5 Events in Multi-Channel Signals: Definition, Examples, and Problem Statement . 49  

6 Cost Definition, Computation, and Minimization . 53  
6.1 Exponentially-Weighted Squared Error Cost . 53  
6.1.1 Cost Definition . 53  
6.1.2 Recursive Computation of the Cost Function . 54  
6.1.3 Cost Function Minimization . 56  
6.2 LSSM-Weighted Squared Error Cost . 66  
6.2.1 Cost Definition . 66  
6.2.2 Recursive Computation of the Cost Function . 67  
6.2.3 Cost Function Minimization . 70  
6.2.4 Locality, Stability, and Steady-State . 70  
6.3 LSSM-Weighted Time-Dependent Polynomial Cost . 72  
6.3.1 Cost Definition . 72  
6.3.2 Recursive Computation of the Cost Function . 73  
6.3.3 Cost Function Minimization . 75
7 Event Detection in Various Conditions

7.1 Principles of Detection

7.1.1 Local Likelihood

7.1.2 Detection as Hypothesis Test

7.2 Detection in the Presence of Interferences

7.2.1 Additive Interference

7.2.2 Multiplicative Interference

7.3 Detection and Classification

7.3.1 Using Multiple Hypothesis Tests

7.3.2 Using a Classifier

7.4 Detection of Model Switches in Model-Dependent Noise

7.4.1 Two-Sided Local Likelihood

7.4.2 Another Family of Hypothesis Tests

7.4.3 Detection of Model Switches

8 Example Applications

8.1 Robust ECG Shape Detection and Separation

8.1.1 LSSM for Measurement Signals

8.1.2 Detecting ECG Signal Shapes

8.2 Detecting and Estimating Straight-Line Movements of a Magnet From Magnetic Field Measurements

8.2.1 Physical Model of the Magnetic Field

8.2.2 LSSM for Magnetic Field Signals

8.2.3 Detecting and Estimating Movements of a Magnet

8.2.4 Results

8.3 Detecting and Estimating Electric Dipole Movements From Electrode Measurements

8.3.1 Physical Model of Voltage Measurements

8.3.2 LSSM for Measurement Signals

8.3.3 Detecting and Estimating Electric Dipole Movements

8.3.4 Results

8.4 Detection and Estimation of a Modulated Signal

8.5 Detection and Classification of Fires

8.5.1 Physical Model of the Optical Smoke Detector

8.5.2 LSSM for the Measured Signals

8.5.3 Detection and Classification of Fires and Nuisances

8.5.4 Results

8.6 Detection and Estimation of Model Switches

8.6.1 Piecewise Lines With Varying Noise Variance
III Learning Sparse Signal Decompositions in a State Space World

9 Why Should We Learn Sparse Signal Decompositions?

10 A State Space World for Sparse Signal Representations

11 Sparsity by Gaussian Scale Mixtures and Type-II Estimation

12 Learning Sparse Signal Decompositions
## 12.4.3 Dual Mean/Dual Precision Parameterization

## 13 Relevant Applications of Learning Sparse Signal Decompositions

### 13.1 Learning Repetitive Signal Shapes
- **Problem Setup**
- **Synthetic Example of Signal Separation and Learning**
- **ECG Signal Separation and Learning**
- **Fetal ECG Signal Separation and Learning**

### 13.2 Learning Classes of Signal Shapes
- **Problem Setup**
- **System Learning**
- **Multi-Sinusoidal System**
- **Model Switches**

### 13.3 Sparse Signal Decomposition in a Signal Shape Domain
- **Problem Setup**
- **Inferring Electric Dipole Movements From Multi-Channel Electrode Measurements**

## 14 Hierarchical Modeling and Processing

### 14.1 Straightforward Hierarchical Processing

### 14.2 A Modeling Layer for the Inputs

### 14.3 A Modeling Layer for the Input Variances
- **General Model**
- **Independent Model**
- **Equality Constraints**
- **Individual Input Constraints**

## 15 Conclusion

## Appendices

### A Matrix Formulas

### B LSSM-Weighted Squared Error and Covariance Estimation
- **Isotropic Uncorrelated Noise**
- **Uncorrelated Noise**
- **Fully Correlated Noise**
List of Symbols

Probability and Statistics

\( \sim \) \quad \text{distributed according to}
\( \overset{iid}{\sim} \) \quad \text{independently and identically distributed according to}
\( \delta(\cdot) \) \quad \text{Dirac delta distribution}
\( \mathcal{N}(m, V) \) \quad \text{multivariate Gaussian distribution of mean } m \text{ and covariance matrix } V
\( \mathcal{N}(x : m, V) \) \quad \frac{1}{\sqrt{(2\pi)^n |V|}} \exp\left(-\frac{1}{2}(x - m)^TV^{-1}(x - m)\right)

Matrix and Vector Operations

\( \|v\| \) \quad L2 norm of a vector \( v \)
\( \langle v, u \rangle \) \quad \text{standard inner product between two vectors } v \text{ and } u
\( \text{tr}(A) \) \quad \text{trace of a matrix } A
\( |A| \) \quad \text{determinant of a squared matrix } A
\( \|A\|_F \) \quad \text{Frobenius norm of a matrix } A \text{ (i.e., } \text{tr}(A^T A))
\( \|A\|_* \) \quad \text{nuclear norm of a matrix } A \text{ (sum of its singular values)}
\( \{A\}_{i,j} \) \quad \text{element } (i, j) \text{ of a matrix } A
\( \{A\}_j \) \quad \text{\( j \)th column of a matrix } A
\( v^{(j)} \) or \( \{v\}_j \) \quad \text{\( j \)th element of a vector } v
\( I_n \) \quad n \times n \text{ identity matrix}
\( I_{m,n} \) \quad m \times n \text{ pseudo identity matrix}
\( 0_{m \times n} \) \quad m \times n \text{ matrix of zeros}
\( 1_{m \times n} \) \quad m \times n \text{ matrix of ones}
\( \text{vec}(A) \) \quad \text{column vector obtained by stacking columns of a matrix } A \text{ on top of one another}
diag($v$) diagonal matrix with diagonal elements being $v$

\[
\text{diag}(A_1, \ldots, A_n)\] block-diagonal matrix with diagonal blocks $A_1, \ldots, A_n$

$\otimes$ Kronecker product

\[
R(\omega) = \begin{bmatrix}
\cos \omega & -\sin \omega \\
\sin \omega & \cos \omega
\end{bmatrix}
\]

\section*{Sets}

\[
N = \{1, 2, 3, \ldots\}
\]

\[
N_0 = \{0, 1, 2, \ldots\}
\]

$\mathbb{R}$ the set of real numbers

$\mathbb{R}_+$ the set of non-negative real numbers

$\mathbb{R}^*$ the set of non-zero real numbers

$\mathbb{R}^*_+$ the set of strictly positive real numbers

$1 = \{1, 1\}$

$S^3 = \{u \in \mathbb{R}^3 : \|u\| = 1\}$

$O_n(\mathbb{R}) = \{U \in \mathbb{R}^{n \times n} : U^TU = I_n\}$

$SO_n(\mathbb{R}) = \{U \in \mathbb{R}^{n \times n} : U^TU = I_n, |U| = 1\}$

$S_+$ the set of symmetric positive semi-definite matrices (of given size)

$S^*_+$ the set of symmetric positive definite matrices (of given size)

\section*{Acronyms}

ARD automatic relevance determination

BSS blind source separation

DARE discrete algebraic Riccati equation

ECG electrocardiogram

EM expectation maximization

GSM Gaussian scale mixture

LLR log-likelihood ratio

LSSM linear state space model

MAP maximum a posteriori

NUV normal with unknown variance

PCA principal component analysis

SBL sparse Bayesian learning

(W)OPP (weighted) orthogonal procrustes problem
Chapter 1

Introduction

Signals can often be labeled with a small set of events such that each event is categorized according to its surrounding signal shapes. The goal of this thesis and many signal processing tasks is to detect and estimate a set of events that are likely to explain the observed signals when fed to a suitably-estimated signal generator, following a causal relationship. Typically, events belonging to the same category trigger the same component of the signal generator, which produces signal shapes with shared characteristics.

This concept is naturally illustrated with the example of an abdominal electrocardiogram (ECG) recording of a pregnant woman. The heart beats of the mother and the fetus can be interpreted as events. While all maternal heart beats produce very similar signal shapes of high amplitude, all fetal heart beats also generate their characteristic signal shapes of lower amplitude. In this example, given an ECG recording of a pregnant woman, the ambition is to detect all maternal and fetal heart beats, estimate the maternal and fetal ECG signal shapes, and appropriately classify each heart beat. This also implies separating the maternal and fetal signals.

Unlike independent and identically distributed samples, signals exhibit correlation and resemblance in time and within measurement channels (for multi-channel signals). Resemblance is to be understood as occurrences of signal shapes with common properties. Hence, exploiting temporal and spatial information in order to estimate, find, and classify signal shapes with shared characteristics is at the essence of signal processing. Many signal processing algorithms, such as the popular matched
filter or a projection onto some signal basis, have been devoted to bring to light temporal and spatial resemblances while keeping an affordable computational complexity. Nevertheless, they often come with substantial limitations and easily become obsolete as soon as assumptions (e.g., the knowledge of a predefined signal shape or a useful signal basis) are violated.

In this thesis, we wish to extract temporal and spatial information from single-channel and multi-channel discrete-time signals by learning sparse signal decompositions. The central idea is to learn signal components such that each component accounts for a characteristic resemblance that is described by a sparse representation in some unknown signal space. Thus, at the same time, we want to separate signal components, estimate signal shapes, find sparse signal representations, and learn signal spaces in which the signals can be sparsely represented. Note that for each of these problems many methods exist but very few of them consider all the four problems jointly in a unified formulation (neglecting straightforward approaches that directly iterate between other standard methods).

In order to exploit temporal and spatial resemblances, we use a powerful signal representation provided by linear state space models (LSSMs). Such models describe a signal by a projection of a state trajectory where the states evolve with time according to a linear first-order difference equation. They have very attractive properties. First, LSSMs can generate quite complex signals with a representation involving few parameters only. In particular, with a high enough state space order, we can model a given signal with any wanted accuracy. Secondly, the time description of a LSSM naturally includes all time-shifted versions of a signal shape without extra effort. Third, the time structure along with the latent state variables allow the use of efficient learning algorithms such as message passing and expectation maximization (EM). Last but not least, LSSMs have vast and flexible modeling capabilities which are particularly suited to model physical signals and handle signal constraints. In a nutshell, LSSMs fundamentally act as a regularization by essentially choosing a signal class and a measure of fit while preserving the natural time structure of discrete-time signals.

In this thesis, we develop a general approach to decompose a (multi-channel or single-channel) discrete-time signal into signal components, where each component is assumed to have a sparse representation in the input domain of its own, known or unknown, LSSM. While the LSSM of a signal component is seen as a signal generator, the non-zero inputs to
the LSSM of a signal component are interpreted as events. The charm of the method resides in its flexibility towards the level of knowledge (i.e., known, unknown, or partially known), which allows a unifying view on the three problems: signal separation, finding sparse signal representations, and learning signal spaces in which the signals can be sparsely represented.

1.1 Outline and Contribution

This thesis is organized in three parts. While Part I summarizes standard properties of LSSMs, Parts II and III contain most of the contribution of this thesis and highly rely on Part I.

Part I: Linear State Space Models

In Part I, which consists of Chapters 2 to 4, we recall important properties of LSSMs. First, Chapter 2 summarizes general facts about LSSMs under the Gaussian assumption.

Secondly, Chapter 3 contains an extensive overview on autonomous LSSMs, which are a special class of LSSMs. The content of Chapter 3 is not novel but yet deals with many interesting aspects of autonomous LSSMs among which some might have been overlooked in the literature. In particular, we show that the set of autonomous LSSM signals is a ring and that the correlation function between a LSSM signal and any discrete-time signal can be recursively and efficiently computed. As a matter of fact, Chapter 3 contains crucial properties that are essential for the methods described in Part II and, to a certain extent, also in Part III.

Finally, given a LSSM under the Gaussian assumption, it is well known that all variables are Gaussian random vectors and that the computation of the posterior densities and related quantities can be done efficiently using Gaussian message passing algorithms [45]. Since message passing algorithms are a topic in its own and have already been extensively studied (e.g., see [11,44,45,61]), Chapter 4 recalls one message passing algorithm, namely, the modified Bryson-Frazier smoother and provides the relations between messages and quantities that are needed in the algorithms of Part III.
Part II: Detection and Estimation in Multi-Channel Signals With Autonomous Linear State Space Models

In Part II, which contains Chapters 5 to 8, we consider a multi-channel (or single-channel) discrete-time signal which can be interpreted as the result of successive events, where each event locally creates characteristic signal shapes. In addition, events are assumed to be separated enough across time such that the local estimation of the signals produced by one specific event is barely influenced by the effects of other events. The general method proposed in Part II aims to detect such events and estimate parameters of these events.

As developed in Chapter 5, we model the signals produced by an event with a two-sided autonomous LSSM signal where the observation matrix encompasses the event parameters to be estimated. The right-sided LSSM accounts for the observed signals after an event while the left-sided model accounts for the observed signals before an event. Thus, a class of events is mapped to a two-sided autonomous LSSM with a set of allowed observation matrices.

Then, detecting and estimating an event amounts to locally fitting an autonomous LSSM to discrete-time observations at any given time. For this purpose, we introduce three local cost functions in Chapter 6 which successively generalize one after another. The attractive property of those cost functions is that they can all be efficiently computed thanks to recursive expressions. Moreover, since two of the cost functions are general quadratic forms, minimizing those costs is a constrained quadratic optimization problem for which closed-form solutions or low-complexity iterative optimization algorithms are often available.

Based on those cost functions, we further define several log-likelihood ratios in Chapter 7 for event detection. Using an adequate notion of local likelihood, we derive multiple hypothesis tests that can handle detection in various conditions, such as in the presence of an unknown additive or multiplicative interference signal.

Finally, Chapter 8 provides an assortment of applications where the proposed method for event detection and estimation reveals its potential.

Part III: Learning Sparse Signal Decompositions in a State Space World

In Part III, which comprises Chapters 9 to 14, we develop a general approach to learn sparse signal decompositions.
After motivating the problem of learning sparse signal decompositions in Chapter 9, Chapter 10 advocates the use of LSSMs driven by both white Gaussian noise and sparse inputs in order to model signal components. Despite its natural simplicity, this representation reveals to be quite flexible in modeling many physical signal components, such as a signal consisting of repetitive signal shapes or a wandering baseline.

In Chapter 11, we introduce sparsity by modeling inputs with independent zero-mean normal variables with unknown variances (NUV), as in the sparse Bayesian learning framework. We show that this type of regularization not only brings sparsity to the signal component LSSMs but also regularizes the estimation of the LSSM parameters.

Then, by combining the signal model of Chapter 10 with the sparsity regularization of Chapter 11, learning sparse signal decompositions boils down to estimating the LSSM parameters and the input variances by maximum likelihood, as explained in Chapter 12. We also derive an efficient expectation maximization (EM) algorithm and other variants to perform such estimation. We demonstrate that all parameters can be jointly updated with closed-form expressions and that all expectation quantities are efficiently computed using a Gaussian message passing algorithm.

Chapter 13 presents several relevant applications with practical and simulated examples. We address the problems of learning repetitive signal shapes, learning classes of signal shapes, and decomposing a signal with scaled, time-shifted, and time-dilated versions of a signal shape. Among all those examples, the fetal ECG signal separation is one of the most promising ones that reflects the power of the proposed approach.

Finally, in Chapter 14, we outline ideas on how to handle more structures in the input domain of a LSSM in order to, for instance, detect synchronous signal components.

1.2 Related Work

Linear State Space Models

Linear state space models along with Kalman filters have long been used in control system \textsuperscript{[3,35,37,83]}. In machine learning and signal processing, Gaussian linear state space models have become increasingly popular \textsuperscript{[9,26,44,45,61,65]} and are still a topic of research.
Event Detection and Estimation

Several approaches have been proposed for event detection and estimation and few important references are [38, 39, 58, 59]. Another original approach consists of using switching dynamical models [28].

In this thesis, we generalize the approach proposed in [46, 61].

Learning Sparse Signal Decompositions

The problem of learning sparse signal decompositions can be seen as a combination of three well-known problems: system identification, compressed sensing, and blind source separation.

System identification [43] essentially consists in estimating an input/output relation given training samples. A known input is fed to an unknown system. Based on the observation of the outputs, one wants to estimate the system dynamic.

Compressed sensing [15] aims to reconstruct a sparse vector given a measurement vector and a dictionary (i.e., a system matrix). In order to enforce sparsity, regularization is used, among which $L1$ regularization is the most popular [15]. In addition, many reconstruction algorithms have been proposed, such as matching pursuit [51] and iterative reweighted least squares minimization [22]. Another way to achieve sparsity is sparse Bayesian learning [69, 81], which is the technique we use in this thesis.

Blind source separation (BSS) [19, 20] focuses on estimating a set of source signals based on the observation of several measurement channels, each of them mixing the signal sources in its own way. Note that both the source signals and the mixing effect need to be estimated. Typically, a statistical model is assumed for the source signals [16] while the mixing effect is often considered to be linear or convolutive [68]. In addition, when source signals are assumed to be sparsely represented in some known domain, many algorithms [5, 27, 93] have been proposed for separating and estimating the source signals. We already see that a system identification problem is somehow a special case of a BSS problem.

Finally, our problem of learning sparse signal decompositions can be seen from two main points of view. Namely, it can be seen as a compressed sensing problem with an unknown dictionary (which thus becomes a dictionary learning problem) or as a BSS problem where source signals have a sparse representation in some unknown signal space. As a consequence, several algorithms have been proposed to learn sparse signal decompositions [2, 42, 55, 93] and usually iterate between finding
sparse sources given a signal space and learning a signal space given sparse sources.

The approach proposed in Part III is based on a preliminary work [11, 13] which deals with a blind deconvolution problem and more precisely, a system identification problem with unknown sparse inputs. Additionally, methods to combine sparsity with temporal and spatial correlations have also been proposed in [60, 91, 92].

In this thesis, we provide a general approach based on LSSM to tackle the problem of learning sparse signal decompositions. Unlike standard learning algorithms (dictionary learning or BSS), our approach exhibits the following advantages:

- the dictionary and the sparse sources can be jointly updated,
- each iteration of the EM learning algorithm has a linear complexity with respect to the number of signal samples,
- the number of sources need not be smaller than the number of measurement channels,
- temporal and spatial correlations are naturally handled by the LSSM which offers a detailed modeling of signals,
- no need to further regularize the dictionary since the LSSM already acts as a regularizer,
- known or constrained signal components are easily dealt with,
- a wandering baseline signal is also seen as a sparse signal component.

We also point out that the sparsity assumption for modeling signals or signal components and for regularizing signal reconstruction problems has been advocated by many authors such as in [10, 71, 90, 93].

1.3 Acknowledgments

This thesis contains some of the plots and simulation results obtained within the scope of several master theses under my supervision. Most of those plots and simulation results are already published in a conference proceeding. Thus, the following contributors need to be properly acknowledged: Christian Käslin [84], for Sec. 8.2, Désirée Clausen [86],
for Sec. 8.3, Elizabeth Ren, for Sec. 8.5, Raphael Keusch, for Sec. 13.1, and Timon Künzle, for Sec. 13.3.2. We also thank Clément Luneau and Andreas Büchel for valuable insights.

Last but not least, we acknowledge Reto A. Wildhaber for inspiration and valuable feedback on Part II and Hampus Malmberg for inspiration and valuable feedback on Part III.

1.4 Notation

Most of the notation is introduced all along the thesis when needed. For the basic symbols, see the “List of Symbols”.

Unlike standard notation in machine learning literature, the probability density function of a multivariate Gaussian distribution of mean \(m \in \mathbb{R}^n\) and symmetric positive definite covariance matrix \(V \in \mathbb{R}^{n \times n}\) is denoted

\[
\mathcal{N}(x : m, V) = \frac{1}{\sqrt{(2\pi)^n|V|}} \exp \left( -\frac{1}{2} (x - m)^\top V^{-1} (x - m) \right),
\]

for \(x \in \mathbb{R}^n\).

For the factor graph notation, we use the same convention as [45].
Part I

Linear State Space Models
“Le cœur a ses raisons que la raison ne connaît point.”
“The heart has its reasons which reason knows nothing of.”

_Pensées_, Blaise Pascal
Chapter 2

Generalities on Linear State Space Models

Let $M \in \mathbb{N}$. We wish to explain $M$-channel observations $y_1, \ldots, y_K \in \mathbb{R}^M$ of duration $K \in \mathbb{N}$ where $K$ is large, as the output of a LSSM of order $n \in \mathbb{N}$ with states $X_k \in \mathbb{R}^n$ given by

\[
\begin{cases}
X_k &= AX_{k-1} + B_k U_k + E_k \\
y_k &= CX_k + Z_k,
\end{cases}
\tag{2.1}
\]

for $k \in \{1, \ldots, K\}$, with state transition matrix $A \in \mathbb{R}^{n \times n}$, observation matrix $C \in \mathbb{R}^{M \times n}$, input matrices $B_k \in \mathbb{R}^{n \times J}$ (with $J \in \mathbb{N}$), and variables $X_0 \in \mathbb{R}^n$, $Z_k \in \mathbb{R}^M$, $U_k \in \mathbb{R}^J$, and $E_k \in \mathbb{R}^n$. Unlike standard representations, the input to the LSSM is decomposed into two parts: $U_k$ and $E_k$. Furthermore, $B_k$ might depend on index $k$.

Such a representation becomes meaningful as soon as we introduce structures, constraints, and statistical assumptions on the LSSM parameters and variables. Note that without restricting parameters and variables, the representation in (2.1) can exactly explain the observations (over-fitting), which is essentially of no use. We now provide some properties and characterizations of such models.
2.1 Solving the LSSM Equations

Solving the system of equations (2.1), $y = \text{vec}(y_1, \ldots, y_K)$ is obtained as the output of the linear system

$$y = TX_0 + HU + PE + Z,$$

where $U = \text{vec}(U_1, \ldots, U_K)$ (and analogously for $E$ and $Z$) and with the transfer matrices

$$T = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^K \end{bmatrix} \in \mathbb{R}^{(KM) \times n} \quad (2.3)$$

$$H = \begin{bmatrix} CB_1 & 0 & \cdots & 0 \\ CAB_1 & CB_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ CA^{K-1}B_1 & CA^{K-2}B_1 & \cdots & CB_K \end{bmatrix} \in \mathbb{R}^{(KM) \times (KJ)} \quad (2.4)$$

$$P = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{K-1} \end{bmatrix} \begin{bmatrix} 0 \\ C \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(KM) \times (Kn)} \quad (2.5)$$

The matrix $P$ is block-Toeplitz. The matrix $H$ is often referred as the system matrix and is not block-Toeplitz unless $B_k = B$ for all $k$.

2.2 Statistical Assumptions

Making a statistical assumption on the variables $X_0$, $Z_k$, $U_k$, and $E_k$, $k \in \{1, \ldots, K\}$ (i.e., considering them as random vector variables), is equivalent to defining a joint probability density function $p(y, x, u, e, z | \theta)$, where $\theta$ is the set of all LSSM parameters and all parameters of the prior distributions. If we assume that the variables $X_0$, $Z_k$, $U_k$, and $E_k$, $k \in \{1, \ldots, K\}$, are mutually independent, then the joint density factorizes into

$$p(y, x, u, e, z | \theta) = p(x_0 | \theta) \prod_{k=1}^{K} \delta(y_k - (Cx_k + z_k))p(z_k | \theta)$$

$$\cdot \delta(x_k - (Ax_{k-1} + B_ku_k + e_k))p(e_k | \theta)p(u_k | \theta), \quad (2.6)$$
which can be represented with a factor graph \[45\].

In this thesis, we assume that \(X_0, Z_k, U_k, \text{ and } E_k, k \in \{1, \ldots, K\}\), are either independent Gaussian random vectors or deterministic quantities (which can be seen as a degenerate Gaussian random vector with a zero covariance matrix).

We often assume the measurement noise \(Z_k \in \mathbb{R}^M\) to be white Gaussian noise, uncorrelated, and of equal variance \(\sigma_Z^2 > 0\) over the measurement channels, i.e.,

\[
Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2 I_M).
\] (2.7)

Thus, we obtain the conditional density

\[
p(y|x_0, u, e, \theta) = \mathcal{N}(y : Tx_0 + Hu + Pe, \sigma_Z^2 I_{MK}).
\] (2.8)

Concerning the variable \(X_0 \in \mathbb{R}^n\), we either consider it as a deterministic unknown quantity \((X_0 = x_0)\) or as deterministically zero. When \(X_0 = x_0\) is deterministic but unknown, we further assume that the LSSM as in (2.1) has no inputs, i.e., \(U_k = 0\) and \(E_k = 0\), for all \(k\). Such a LSSM is referred as an autonomous LSSM whose likelihood function is

\[
p(y|x_0, \theta) = \mathcal{N}(y : Tx_0, \sigma_Z^2 I_{MK})
\] (2.9)

\[= \prod_{k=1}^{K} \mathcal{N}(y_k : CA^k x_0, \sigma_Z^2 I_M)
\] (2.10)
\[ y_k - C A^k x_0 \right)^2 \right). \quad (2.11) \]

Autonomous LSSMs are further investigated in Chapter 3.

When \( X_0 \) is deterministically zero, inputs are used (either the \( U_k \)'s or the \( E_k \)'s or both). The random vector \( E_k \in \mathbb{R}^n \) is referred as the state noise variable and is always assumed to be zero-mean Gaussian with a time-invariant covariance matrix \( V_E \in \mathbb{S}^* \), i.e.,

\[
E_k \overset{iid}{\sim} \mathcal{N}(0, V_E). \quad (2.12)
\]

Thus, we get the conditional density

\[
p(y|u, \theta) = \int p(y|u, e, \theta) p(e|\theta) \, de
\]

\[
= \mathcal{N} \left( y : Hu, \sigma^2_Z I_{MK} + PV_E P^T \right), \quad (2.14)
\]

with

\[
V_E = \text{diag}(V_E, \ldots, V_E) \in \mathbb{R}^{(Kn) \times (Kn)}. \quad (2.15)
\]

Note that the state noise effectively acts as correlated (in time) observation noise. When fitting the signal vector \( Hu \) to \( y \), the covariance \( V_E \) modifies the standard \( L^2 \) norm. In particular, when there is no state noise (i.e., \( V_E = 0 \)), we have

\[
p(y|u, \theta) = \mathcal{N}(y : Hu, \sigma^2_Z I_{MK}), \quad (2.16)
\]

which coincides with measuring the data fit with the standard \( L^2 \) norm.

The random vector \( U_k \in \mathbb{R}^J \) is referred as the input to the LSSM. We assume that

\[
U_k \sim \mathcal{N}(m_U, \Sigma_U). \quad (2.17)
\]

Thus, the likelihood is

\[
p(y|\theta) = \mathcal{N}(y : Hm_U, \sigma^2_Z I_{MK} + H\Sigma_U H^T + PV_E P^T), \quad (2.18)
\]

with

\[
\Sigma_U = \text{diag}(\Sigma_U_1, \ldots, \Sigma_U_K) \quad (2.19)
\]

\[
m_U = \text{vec}(m_U_1, \ldots, m_U_K). \quad (2.20)
\]
Actually, our input model is quite often zero-mean, which leads to
\[ p(y|\theta) = N(y : 0, \sigma^2 I_{MK} + H\Sigma_U H^T + PV_E P^T). \] (2.21)

The joint density (2.6) of this Gaussian LSSM can be represented with the factor graph of Fig. 2.1.

Despite its simplicity, the Gaussian assumption is flexible enough to derive powerful and efficient signal processing algorithms. Remarkably, sparse signal estimation can still be achieved within the Gaussian assumption when some of the variance parameters are unknown, as explained in Chapter 11. Besides, the use of Gaussian LSSM can be seen as an approach to regularize the covariance estimation problem. Indeed, the LSSM implies a structure on the covariance matrix of the data
\[ V_{\text{tot}} = \sigma^2 I_{MK} + H\Sigma_U H^T + PV_E P^T, \] (2.22)
as shown in (2.18). From the point of view of fitting a signal model, LSSMs are essentially a way to restrict the class of signals (in particular via the transform matrix \( H \)) and to choose a norm for measuring data fit, as shown in (2.14).

### 2.3 LSSM Inputs Fed to a LSSM

Consider a LSSM model as in (2.1). In addition, assume that the input \( U_k \in \mathbb{R}^J \) is also generated with a LSSM written as
\[
\begin{align*}
\tilde{X}_k &= \tilde{A}\tilde{X}_{k-1} + \tilde{B}_k \tilde{U}_k + \tilde{E}_k \\
U_k &= \tilde{C}\tilde{X}_{k-1} + \tilde{Z}_k,
\end{align*}
\] (2.23)
with suitably defined parameters and variables. Note that the second equation in (2.23) is slightly different from the one in (2.1) for convenience and without loss of generality.

As a result, the signal \( y \) is explained as the output of a LSSM as in (2.1) but with a time-varying state transition matrix. Indeed, the joint state transition equation can be written as
\[
\begin{bmatrix}
X_k \\
\tilde{X}_k
\end{bmatrix} = 
\begin{bmatrix}
A & B_k\tilde{C} \\
0 & \tilde{A}
\end{bmatrix}
\begin{bmatrix}
X_{k-1} \\
\tilde{X}_{k-1}
\end{bmatrix} + 
\begin{bmatrix}
0 \\
\tilde{B}_k
\end{bmatrix}
\tilde{U}_k + 
\begin{bmatrix}
E_k + B_k\tilde{Z}_k \\
\hat{E}_k
\end{bmatrix}.
\] (2.24)

In particular, if \( B_k\tilde{C} \) is independent of \( k \), then this joint LSSM is again of the form (2.1) (i.e., with a time-invariant state transition matrix).
2.4 LSSM Forward and Backward in Time

2.4.1 LSSM Backward in Time

When inspecting the LSSM introduced in (2.1), we observe that an input $U_k$ at a given index $k$ has a causal effect by only affecting the state variables $X_i$, for $i \geq k$. Such LSSM is qualified as forward in time.

Nonetheless, in signal processing, it is sometimes useful to have an anti-causal effect. Thus, we also define a LSSM backward in time as

$$
\begin{align*}
X_{k-1} &= A(X_k + B_k U_k + E_k) \\
y_k &= CX_k + Z_k,
\end{align*}
$$

(2.25)

$k \in \{1, \ldots, K\}$, of order $n \in \mathbb{N}$ with parameters $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{M \times n}$, $B_k \in \mathbb{R}^{n \times J}$ (with $J \in \mathbb{N}$), and variables $X_K \in \mathbb{R}^n$, $Z_k \in \mathbb{R}^M$, $U_k \in \mathbb{R}^J$, and $E_k \in \mathbb{R}^n$. Unlike the model defined forward in time, the model defined backward in time retains $X_K$ as initial variable and not $X_0$ anymore. Note also the difference to the state trajectory equation. Namely, inputs are added before the multiplication by the state transition matrix $A$, without loss of generality.

2.4.2 LSSM Forward/Backward in Time

Combining a LSSM defined forward in time with a LSSM defined backward in time, we obtain a LSSM forward/backward in time

$$
\begin{align*}
X^f_k &= A^f X^f_{k-1} + B^f_k U_k + E^f_k \\
X^p_{k-1} &= A^p (X^p_k + B^p_k U_k + E^p_k) \\
y_k &= C^f X^f_k + C^p X^p_k + Z_k,
\end{align*}
$$

(2.26)

$k \in \{1, \ldots, K\}$, with suitably defined parameters and variables. The initial conditions are set through the variables $X^f_0$ and $X^p_{K+1}$. The input vector $U_k$ affects both the forward and backward models, which creates a two-sided effect.
Chapter 3

Autonomous Linear State Space Models

3.1 Definitions and Outline

Definition 1 (Multi-channel LSSM signal). Let $M \in \mathbb{N}$. A $M$-channel discrete-time signal $f_k \in \mathbb{R}^M$, $k \in \mathbb{Z}$, is a LSSM signal (i.e., generated by a two-sided autonomous LSSM) if and only if there exists $C_p \in \mathbb{R}^{M \times n_p}$, $A_p \in \mathbb{R}^{n_p \times n_p}$, $s_p \in \mathbb{R}^{n_p}$, and $C_f \in \mathbb{R}^{M \times n_f}$, $A_f \in \mathbb{R}^{n_f \times n_f}$, $s_f \in \mathbb{R}^{n_f}$, for some $(n_p, n_f) \in \mathbb{N}_0^2 \setminus \{0,0\}$ such that

$$f_k = \begin{cases} C_p A_p^{|k|} s_p & \text{for } k \leq 0 \\ C_f A_f^k s_f & \text{for } k > 0. \end{cases} \quad (3.1)$$

We denote $\{C, A, s\}$ the LSSM parameters of such signal, which is to be understood as $C = \{C_p, C_f\}$, $A = \{A_p, A_f\}$, and $s = \{s_p, s_f\}$. We often consider such LSSM signal $f(C)$ as a function of $C$ such that

$$C = [\mathbb{C}] \triangleq [C_p \ C_f] \in \mathbb{R}^{M \times n}, \quad (3.2)$$

with

$$n = n_p + n_f. \quad (3.3)$$

The change point of this two-sided model is defined to be at time $k = 0$. However, this is not a restriction since such a LSSM signal will be shifted by a time of interest. A LSSM signal can also be made left-sided or right-sided by setting $n_f = 0$ or $n_p = 0$. 

17
Definition 2 (LSSM signal). A LSSM signal \( f_k \in \mathbb{R}, k \in \mathbb{Z} \), is a single-channel LSSM signal (i.e., with \( M = 1 \)).

In this chapter, we investigate the LSSM signals, their properties, parameterizations, algebraic structure, and estimation. In particular, we prove that the set of LSSM signals, which is a vector space and a ring, consists of linear combinations of (two-sided) exponentials, polynomials, sinusoids, finite-length signals, and products of those. We often omit the adjectives “autonomous” and “scalar”. All the sections of this chapter concern right-sided scalar LSSM signals. However, the adaptation to the cases of two-sided and multi-channel is straightforward or stated explicitly.

As a matter of taste, when using solely a right-sided LSSM signal (without a left side), we prefer to start indexing the right-sided signal at \( k = 0 \) instead of \( k = 1 \), as opposed to Definition 2. Thus, we have:

Definition 3 (Right-sided LSSM signal). A right-sided discrete-time signal \( f \) is generated by an autonomous LSSM if and only if there exists \( C \in \mathbb{R}^{1 \times n}, A \in \mathbb{R}^{n \times n}, \) and \( s \in \mathbb{R}^n \), for some \( n \in \mathbb{N} \), such that for \( k \geq 0 \),

\[
f_k = CA^k s. \tag{3.4}
\]

The LSSM parameters of a right-sided LSSM signal are denoted \( \{C, A, s\} \) (unlike the two-sided case).

### 3.2 Algebraic Structure of the Set of LSSM Signals

Proposition 1. The set of scalar right-sided LSSM signals is a vector space.

Proof of Proposition 1: Let \( f^{(1)} \) and \( f^{(2)} \) be two right-sided LSSM signals with respective parameters \( \{C_1, A_1, s_1\} \) and \( \{C_2, A_2, s_2\} \). Then, for all \( \lambda \in \mathbb{R} \), for \( k \geq 0 \), we have

\[
f^{(1)}_k + \lambda f^{(2)}_k = C_1 A_1^k s_1 + \lambda C_2 A_2^k s_2 \tag{3.5}
\]

\[
= \begin{bmatrix} C_1 & \lambda C_2 \end{bmatrix} \text{diag}(A_1, A_2)^k \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \tag{3.6}
\]

Thus, \( f^{(1)} + \lambda f^{(2)} \) is also a right-sided LSSM signal with parameters \( \{ \begin{bmatrix} C_1 & \lambda C_2 \end{bmatrix} \text{diag}(A_1, A_2), \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \} \).

Furthermore, we have the following interesting property.
Proposition 2 (Product of right-sided LSSM signals). Let $f^{(1)}$ and $f^{(2)}$ be right-sided LSSM signals with respective parameters $\{C_1, A_1, s_1\}$ and $\{C_2, A_2, s_2\}$. Then, $f^{(1)} \cdot f^{(2)}$ is also a right-sided LSSM signal with parameters $\{C_1 \otimes C_2, A_1 \otimes A_2, s_1 \otimes s_2\}$.

Proof of Proposition 2. For $k \geq 0$, we have

$$f^{(1)}_k \cdot f^{(2)}_k = (C_1 A_1^k s_1)(C_2 A_2^k s_2) \quad (3.7)$$
$$= (C_1 A_1^k s_1) \otimes (C_2 A_2^k s_2) \quad (3.8)$$
$$= (C_1 \otimes C_2)(A_1 \otimes A_2)^k (s_1 \otimes s_2), \quad (3.9)$$

using (A.1), which concludes the proof.

Along with the fact that the constant signal $f = 1$ is a LSSM signal generated with $C = A = s = 1$, the following proposition holds.

Proposition 3. The set of scalar right-sided LSSM signals is a ring.

The analysis for right-sided LSSM signals is easily extended to the case of two-sided LSSM signals, by treating each signal side separately. Thus, we have the following propositions.

Proposition 4 (Product of LSSM signals). Let $f^{(1)}$ and $f^{(2)}$ be LSSM signals with respective parameters $\{C_1, A_1, s_1\}$ and $\{C_2, A_2, s_2\}$. Then, $f^{(1)} \cdot f^{(2)}$ is also a LSSM signal with parameters $\{C_1 \otimes C_2, A_1 \otimes A_2, s_1 \otimes s_2\}$.

We use the convention that the Kronecker product of two (ordered) sets with equal number of elements is the set of the Kronecker products between matching elements. For instance, $C_1 \otimes C_2$ means $\{C_{p,1} \otimes C_{p,2}, C_{f,1} \otimes C_{f,2}\}$. We also denote $1 = \{1, 1\}$ such that the LSSM signal with parameters $C = A = s = 1$ (i.e., the constant unit signal) is the identity element of the ring of LSSM signals.

Proposition 5 (Algebraic structure of LSSM signals). The set of scalar LSSM signals is a vector space and a ring.

These properties will prove to be extremely useful in the remainder of this thesis.

3.3 Equivalent LSSM Signal Form

Proposition 6 (Right-sided LSSM signal form). A right-sided real-valued discrete-time signal $f_k \in \mathbb{R}$, $k \in \mathbb{N}_0$, can be generated by an
autonomous LSSM if and only if it has the form

\[
f_k = h_k + \sum_{q=1}^{Q} \lambda_q \Gamma_q(k) + \sum_{\ell=1}^{L} \rho_{\ell}^k (A_{\ell}(k) \cos(k \omega_{\ell}) + B_{\ell}(k) \sin(k \omega_{\ell})) ,
\]

for \( k \geq 0 \), with \( Q \in \mathbb{N}_0 \), \( L \in \mathbb{N}_0 \),

- \( h_k \in \mathbb{R} \) such that there exists \( \pi_0 \in \mathbb{N}_0 \), such that \( h_k = 0 \) for \( k \geq \pi_0 \) and \( h_{\pi_0-1} \neq 0 \) (i.e., a finite length signal),
- for all \( q \in \{1, \ldots, Q\} \), \( \lambda_q \in \mathbb{R}^* \), \( \Gamma_q \) is a real polynomial in \( k \) of degree \( \pi_q - 1 \), for some \( \pi_q \in \mathbb{N} \),
- for all \( \ell \in \{1, \ldots, L\} \), \( \rho_{\ell} \in \mathbb{R}^*_+ \), \( \omega_{\ell} \in (0, 2\pi) \setminus \{\pi\} \), \( A_{\ell} \) and \( B_{\ell} \) are real polynomials in \( k \), and \( \upsilon_{\ell} \in \mathbb{N} \) is one plus the maximum degree of \( A_{\ell} \) and \( B_{\ell} \).

From Proposition 6, it follows that exponentials, polynomials, and (exponentially decaying) sinusoids are LSSM signals. We now prove Proposition 6.

### 3.3.1 From a Signal Form to a LSSM Parameterization

We here build a LSSM that generates the signal \( f \) in (3.10).

For \( \omega \in \mathbb{R} \), we denote

\[
R(\omega) = \begin{bmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{bmatrix}.
\]

For \( m \in \mathbb{N} \), \( N_m \) denotes the square matrix of dimension \( m \) with ones in the first upper diagonal and zeros elsewhere, i.e.,

\[
N_m = \begin{bmatrix} 0 & 1 \\ & 0 \\ & & \ddots \\ & & & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{m \times m},
\]

for \( m > 1 \), and \( N_1 = 0 \). It follows that \( N_m \) is a nilpotent matrix of order \( m \), i.e., \( N_m^m = 0 \) and \( N_m^p \neq 0 \) for \( p \in \{0, \ldots, m-1\} \). Furthermore, \( N_m^p \) is the matrix containing ones in the \( p \)th upper diagonal and zeros elsewhere.
Before defining our LSSM, we here open a parenthesis on polynomials. Let $B(k)$ be a polynomial of degree $(d - 1) \in \mathbb{N}_0$ and coefficients $b = [b_0 \ldots b_{d-1}]$. We denote

$$k_d = [k^0 k^1 \ldots k^{d-1}]^T,$$

such that $B(k) = bk_d$. As it will become clear later, the LSSM parametrization requires to map the polynomial coefficients from the canonical basis $(1, k, k^2, \ldots)$ to the binomial basis $\left(\binom{k}{0}, \binom{k}{1}, \binom{k}{2}, \ldots\right)$. Indeed, the binomial coefficients appear when computing matrix powers. For all $(i, k) \in \mathbb{N}_0^2$,

$$\binom{k}{i} = \sum_{j=0}^{i} \frac{t_{i,j} \cdot k^j}{j!},$$

where $t_{i,j}$ are the Stirling numbers of the first kind. Thus, for a fixed $i \in \mathbb{N}_0$, $\binom{k}{i}$ is a polynomial in $k$ of degree $i$. Denoting $\tilde{T}_d \in \mathbb{R}^{d \times d}$ the lower triangular matrix such that

$$\{\tilde{T}_d\}_{i+1,j+1} = \begin{cases} t_{i,j}, & \text{for } j \leq i \\ \frac{j!}{j!}, & \text{for } j > i, \end{cases}$$

for $(i, j) \in \{0, \ldots, d-1\}^2$, we have the relation

$$\binom{k}{i} = \{\tilde{T}_d k_d\}_{i+1}.$$

Thus, given the coefficients $b$ of the polynomial $B(k)$ in the canonical basis, its coefficients in the binomial basis are $b \tilde{T}_d^{-1}$. For convenience (for a later use), we introduce $T_d = S_d \tilde{T}_d$, where $S_d$ is the square matrix of dimension $d$ containing ones in the anti-diagonal. Note that $T_d$ is an upper-left triangular matrix such that we have

$$\binom{k}{i} = \{T_d k_d\}_{d-i},$$

for $i \in \{0, \ldots, d-1\}$.

We are now ready to define a LSSM generating the signal $f$ in (3.10). For $q \in \{1, \ldots, Q\}$, we define

$$J_q = \lambda_q (I_{\pi_q} + N_{\pi_q}) \in \mathbb{R}^{\pi_q \times \pi_q}$$

$$s^\circ = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \in \mathbb{R}^{\pi_q}$$

$$C_q = \gamma_q T_{\pi_q}^{-1} \in \mathbb{R}^{1 \times \pi_q},$$
where \( \gamma_q \in \mathbb{R}^{1 \times \pi_q} \) are the coefficients of \( \Gamma_q \). It results that for all \( k \geq 0 \),

\[
C_q v q s_q^o = C_q \left( \lambda_q (I_{\pi_q} + N_{\pi_q}) \right)^k s_q^o
= \lambda_q^k \sum_{i=0}^{k} \binom{k}{i} C_q N_{\pi_q}^i s_q^o
= \lambda_q^k \sum_{i=0}^{\pi_q-1} \{T_{\pi_q} k_{\pi_q}\}_{(\pi_q-i)} \{C_q\}_{(\pi_q-i)}
= \lambda_q^k C_q T_{\pi_q} k_{\pi_q}
= \lambda_q^k \Gamma_q(k),
\]

where we have used the binomial theorem in (3.22) and the relations (3.17) and \( C_q N_{\pi_q}^i s_q^o = \{C_q\}_{(\pi_q-i)} \), \( i \in \{0, \ldots, \pi_q - 1\} \) in (3.23).

For \( \ell \in \{1, \ldots, L\} \), we define

\[
G_\ell = (I_{v_\ell} + N_{v_\ell}) \otimes \rho_\ell R(\omega_\ell) \in \mathbb{R}^{2v_\ell \times 2v_\ell}
\]

and \( \tilde{C}_\ell \) such that

\[
\tilde{C}_{\ell;1} = a_\ell T_{v_\ell}^{-1}
, \quad \tilde{C}_{\ell;2} = b_\ell T_{v_\ell}^{-1}
\]

where \( \tilde{C}_{\ell;1} \) denotes the odd-positioned elements of \( \tilde{C}_\ell \), \( \tilde{C}_{\ell;2} \) the even-positioned elements, \( a_\ell \) are the coefficients of \( A_\ell \), and \( b_\ell \) of \( B_\ell \). It results that for all \( k \geq 0 \), we have

\[
\tilde{C}_\ell G_\ell^k \tilde{s}_\ell^o = \tilde{C}_\ell \left( (I_{v_\ell} + N_{v_\ell}) \otimes \rho_\ell R(\omega_\ell) \right)^k \tilde{s}_\ell^o
= \tilde{C}_\ell \left( (I_{v_\ell} + N_{v_\ell})^k \otimes \rho_\ell^k R(k\omega_\ell) \right) \tilde{s}_\ell^o
= \rho_\ell^k \sum_{i=0}^{v_\ell} \binom{k}{i} \tilde{C}_\ell \left( N_{v_\ell}^i \otimes R(k\omega_\ell) \right) \tilde{s}_\ell^o.
\]

Denoting \( v = [1 \ 0 \ 0 \ 1]^T \), we have the relation

\[
\tilde{C}_\ell (N_{v_\ell}^i \otimes R(k\omega_\ell)) \tilde{s}_\ell^o
= v^T \left( \begin{bmatrix} \tilde{C}_{\ell;1}^i & \otimes & I_2 \end{bmatrix} (N_{v_\ell}^i \otimes R(k\omega_\ell)) \begin{bmatrix} \tilde{s}_\ell^o & \otimes & \tilde{s}_\ell^o \end{bmatrix} \right) v
\]

\[
= v^T \begin{bmatrix} \tilde{C}_{\ell;1} N_{v_\ell}^i \tilde{s}_\ell^o & \tilde{C}_{\ell;2} N_{v_\ell}^i \tilde{s}_\ell^o \end{bmatrix} \otimes R(k\omega_\ell) v
\]
\[\begin{align*}
&= (\tilde{C}_{\ell;1} N_{v\ell} \tilde{s}_{\ell;1}^o + \tilde{C}_{\ell;2} N_{v\ell} \tilde{s}_{\ell;2}^o) \cos(k\omega_{\ell}) \\
&\quad + (\tilde{C}_{\ell;2} N_{v\ell} \tilde{s}_{\ell;1}^o - \tilde{C}_{\ell;1} N_{v\ell} \tilde{s}_{\ell;2}^o) \sin(k\omega_{\ell}),
\end{align*}\]

(3.35)

using (A.1) in (3.33). Furthermore, since \(\tilde{s}_{\ell;2}^o = 0\), we have

\[\begin{align*}
\tilde{C}_{\ell}(N_{v\ell} \otimes R(k\omega_{\ell})) \tilde{s}_{\ell}^o &= \tilde{C}_{\ell;1} N_{v\ell} \tilde{s}_{\ell;1}^o \cos(k\omega_{\ell}) + \tilde{C}_{\ell;2} N_{v\ell} \tilde{s}_{\ell;1}^o \sin(k\omega_{\ell}).
\end{align*}\]

(3.36)

Plugging this expression in (3.32), we finally obtain

\[\begin{align*}
\tilde{C}_{\ell} G_k^s \tilde{s}_{\ell}^o &= \rho_k^k \sum_{i=0}^{v_{\ell}} \binom{k}{i} \tilde{C}_{\ell;1} N_{v\ell} \tilde{s}_{\ell;1}^o \cos(k\omega_{\ell}) + \tilde{C}_{\ell;2} N_{v\ell} \tilde{s}_{\ell;1}^o \sin(k\omega_{\ell})
\end{align*}\]

(3.37)

\[\begin{align*}
&= \rho_k^k (A_k(k) \cos(k\omega_{\ell}) + B_k(k) \sin(k\omega_{\ell})),
\end{align*}\]

(3.38)

where we have used the exact same steps as in (3.22)–(3.25).

For generating the signal \(h_k\) of finite length \(\pi_{0}\), we introduce

\[\begin{align*}
J_0 &= N_{\pi_0} \in \mathbb{R}^{\pi_0 \times \pi_0},
\end{align*}\]

(3.39)

\[\begin{align*}
s_0^o &= [0 \cdots 0 1]^T \in \mathbb{R}^{\pi_0},
\end{align*}\]

(3.40)

\[\begin{align*}
C_0 &= [h_{\pi_0-1} \ h_{\pi_0-2} \ \cdots \ h_0] \in \mathbb{R}^{1 \times \pi_0}.
\end{align*}\]

(3.41)

It results that for \(k \in \{0, \ldots, \pi_0 - 1\},

\[C_0 J_0^k s_0^o = h_k,
\]

(3.42)

and \(C_0 J_0^k s_0^o = 0\), for \(k \geq \pi_0\).

We now stack all those LSSMs into a single one of order

\[n = \pi_0 + \sum_{q=1}^{Q} \pi_q + 2 \sum_{\ell=1}^{L} v_{\ell},
\]

(3.43)

and with parameters

\[\begin{align*}
C &= [C_0 \ C_1 \ \cdots \ C_Q \ \tilde{C}_1 \ \cdots \ \tilde{C}_L] \in \mathbb{R}^{1 \times n},
\end{align*}\]

(3.44)

\[\begin{align*}
A &= \text{diag}(J_0, J_1, \ldots, J_Q, G_1, \ldots, G_L) \in \mathbb{R}^{n \times n},
\end{align*}\]

(3.45)

\[\begin{align*}
s_o &= [(s_0^o)^T \ \cdots \ (s_Q^o)^T \ (\tilde{s}_1^o)^T \ \cdots \ (\tilde{s}_L^o)^T]^T \in \mathbb{R}^n.
\end{align*}\]

(3.46)

Thus, for \(k \geq 0\), we have

\[\begin{align*}
CA^k s_o = C_0 J_0^k s_0^o + \sum_{q=1}^{Q} C_q J_q^k s_q^o + \sum_{\ell=1}^{L} \tilde{C}_\ell G_\ell^k \tilde{s}_\ell^o = f_k,
\end{align*}\]

(3.47)
by combining (3.25), (3.38), and (3.42). The parametrization \( \{C, A, s_0\} \) described in (3.44)–(3.46) is referred as the modified Jordan canonical form. Obviously, the LSSM parameters \( \{C, A, s_0\} \) such that (3.4) holds are not unique. However, the chosen ones allow a straightforward relation with the signal form in (3.10).

3.3.2 From a LSSM Parametrization to a Signal Form

Given a right-sided LSSM signal \( f \) with parameters \( \{C, A, s\} \), we decompose \( A \) in Jordan canonical form

\[
A = P \text{diag}(J_1, \ldots, J_Q) P^{-1},
\]

(3.48)

where \( J_q \) denotes a Jordan block corresponding to either a real eigenvalue or a pair of complex conjugate eigenvalues. Thus, for all \( k \geq 0 \),

\[
CA^k s = \sum_{q=1}^{Q} \tilde{C}_q J_q^k \tilde{s}_q,
\]

(3.49)

where we decompose \( \tilde{C} = CP \) and \( \tilde{s} = P^{-1} s \) in blocks matching the structure of the Jordan decomposition. We now focus on discrete-time signals \( (CJ^k s)_{k \geq 0} \) generated by a Jordan block.

Let \( J \in \mathbb{R}^{\pi \times \pi} \) be a Jordan block corresponding to a real eigenvalue \( \lambda \neq 0 \). Then,

\[
J = \lambda I_\pi + N_\pi.
\]

(3.50)

Using the binomial theorem, it follows that for all \( k \geq 0 \),

\[
CJ^k s = \sum_{i=0}^{\pi-1} \binom{k}{i} \lambda^{k-i} CN_i^\pi s
\]

(3.51)

\[
= \lambda^k \Gamma_\pi(k),
\]

(3.52)

where \( \Gamma_\pi(k) \) is a polynomial in \( k \) of degree \( \pi - 1 \) since for a given \( i \in \mathbb{N}_0 \), \( \binom{k}{i} \) is a polynomial in \( k \) of degree \( i \).

Let \( J \in \mathbb{R}^{\pi \times \pi} \) be a Jordan block corresponding to a zero eigenvalue. Then,

\[
J = N_\pi.
\]

(3.53)
Since \( N_\pi = 0 \), for all \( k \geq \pi \),

\[
C J^k s = 0. \tag{3.54}
\]

Thus, \((C J^k s)_{k \geq 0}\) is a right-sided discrete-time signal of maximum duration \( \pi \).

Let \( J \in \mathbb{R}^{2\pi \times 2\pi} \) be a Jordan block corresponding to a pair of complex conjugate eigenvalues \( \rho e^{\pm i\omega} \), \( \rho \in \mathbb{R}^*_+ \), \( \omega \in (0, 2\pi) \setminus \{\pi\} \). Then,

\[
J = I_\pi \otimes \rho R(\omega) + N_\pi \otimes I_2. \tag{3.55}
\]

Using the binomial theorem, it follows that for all \( k \geq 0 \),

\[
C J^k s = C \sum_{i=0}^{k} \binom{k}{i} (I_\pi \otimes \rho R(\omega))^{k-i} (N_\pi \otimes I_2)^i s \tag{3.56}
\]

\[
= \sum_{i=0}^{\pi-1} \binom{k}{i} C (N_\pi^i \otimes \rho^{k-i} R((k-i)\omega)) s \tag{3.57}
\]

\[
= \rho^k (A_\pi(k) \cos(\omega k) + B_\pi(k) \sin(\omega k)), \tag{3.58}
\]

where \( A_\pi(k) \) and \( B_\pi(k) \) are polynomials in \( k \) of degree \( \pi - 1 \).

Thus, plugging (3.52), (3.54), and (3.58) appropriately in (3.49), the signal \( f \) has the form (3.10).

Referring to Proposition 6, this analysis also demonstrates that \( \lambda_q \in \mathbb{R}^*, q \in \{1, \ldots, Q\} \) are the real (non-zero) eigenvalues of \( A \) of order \( \pi_q \) while \( \rho_\ell e^{\pm i\omega_\ell} \in \mathbb{C}, \rho_\ell \in \mathbb{R}^*_+, \omega_\ell \in (0, 2\pi) \setminus \{\pi\}, \ell \in \{1, \ldots, L\} \) are the pairs of complex conjugate eigenvalues of \( A \) of order \( \nu_\ell \).

## 3.4 Uniqueness of LSSM Signal Form

**Lemma 1** (Unique Characterization). Given the discrete-time LSSM signal \( f \) as in (3.10), with the following extra properties

1. all the \( \lambda_q, q \in \{1, \ldots, Q\} \) are pairwise distinct,
2. all the \( (\rho_\ell, \omega_\ell), \ell \in \{1, \ldots, L\} \) are pairwise distinct,

it results that the parameters

- \( \pi_0, h_k, k = 0, \ldots, \pi_0 - 1 \),
- \( \pi_q, \lambda_q \neq 0 \), polynomials \( \Gamma_q, q \in \{1, \ldots, Q\} \),
• \( \pi_\ell, \rho_\ell > 0, \omega_\ell \in (0, 2\pi) \setminus \{\pi\} \), polynomials \( A_\ell \) and \( B_\ell \) \( \ell \in \{1, \ldots, L\} \) are unique (up to a permutation of indices).

Note that if the extra properties are not satisfied, we can simply merge the terms associated to a same \( \lambda_q \) or a same \( (\rho_\ell, \omega_\ell) \) in (3.10). In terms of the matrix \( A \), those two extra properties mean that all eigenvalues of \( A \) have a geometric multiplicity of 1. Thus, in the Jordan decomposition, each block size corresponds to the algebraic multiplicity of its associated eigenvalue.

When building the LSSM corresponding to \( f \) as in Sec. 3.3.1 with the extra properties, it follows that \( \{A, s_q\} \) is controllable. Indeed, the matrix \( A \) that we obtained is in modified Jordan canonical form and all eigenvalues have a geometric multiplicity of one, i.e., no blocks share the same eigenvalue. Since the last component of \( s_q^o \) is non-zero, for \( q \in \{1, \ldots, Q\} \), and at least one of the last two components of \( \tilde{s}_\ell^o \) is non-zero, for \( \ell \in \{1, \ldots, L\} \), it follows that \( \{A, s_q\} \) is controllable (cf., [83, p. 511], [17]).

Proof of Lemma [1]: Assume that there are two representations of \( f \) as in (3.10). We denote \( g \) the signal of the difference of the two signals which is zero and also has a representation as in (3.10) with the extra property of having pairwise distinct eigenvalues. Building an associated LSSM for \( g \) as in Sec. 3.3.1 we have

\[
g_k = C_g A^k g s^o_g = 0,
\]

for all \( k \geq 0 \). Since \( \{A_g, s^o_g\} \) is controllable by construction, it follows that \( C_g = 0 \). Then, all polynomials in the decomposition of \( g \) as in (3.10) are zero (due to the one-to-one mapping between the observation vector and polynomial coefficients). However, each polynomial in the decomposition of \( g \) as in (3.10) is either a polynomial from the first representation of \( f \), or a polynomial from the second representation of \( f \), or a difference of polynomials between the first and second representations. Since a polynomial in the first or second representation of \( f \) cannot be zero by definition, it implies that each polynomial in the decomposition of \( g \) is a difference of polynomials between the first and second representations of \( f \). This implies that the values \( \lambda_q, \rho_\ell, \omega_\ell, \) \( q \in \{1, \ldots, Q\}, \ell \in \{1, \ldots, L\} \), coincide (up to a permutation of indices) in both representations of \( f \) and that the polynomials and finite-duration signal also coincide.
3.5 Initial State Versus Observation Vector

Given a matrix $A \in \mathbb{R}^{n \times n}$, we are interested in the LSSM signals obtained by varying the parameters $C$ and $s$. In this section, we demonstrate that it is enough to vary either $C$ only or $s$ only while spanning the same set of signals.

**Proposition 7.** Given $\{C, A, s\} \in \mathbb{R}^{1 \times n} \times \mathbb{R}^{n \times n} \times \mathbb{R}^n$, there exists $s_0 \in \mathbb{R}^n$ independent of $C$ and $s$, and $C' \in \mathbb{R}^{1 \times n}$ such that

$$CA^k s = C'A^k s_0, \text{ for all } k \geq 0. \quad (3.60)$$

**Proof of Proposition 7:** We first prove Proposition 7 for matrices $A \in \mathbb{R}^{n \times n}$ for which there exists $s_0 \in \mathbb{R}^n$ such that $\{A, s_0\}$ is controllable.

As a consequence of the Cayley-Hamilton theorem, for all $k \geq 0$, there exists real coefficients $\beta_{k,0}, \ldots, \beta_{k,n-1}$ such that

$$A^k = \sum_{i=0}^{n-1} \beta_{k,i} A^i. \quad (3.61)$$

It follows that for any $(\tilde{C}, \tilde{s}) \in \mathbb{R}^{1 \times n} \times \mathbb{R}^n$,

$$\tilde{C} A^k \tilde{s} = \sum_{i=0}^{n-1} \beta_{k,i} \tilde{C} A^i \tilde{s} \quad (3.62)$$

depends on (at most) the $n$ coefficients $\tilde{C} A^i \tilde{s}$, $i \in \{0, \ldots, n-1\}$. Thus, if

$$C' \begin{bmatrix} s_0 & As_0 & \ldots & A^{n-1}s_0 \end{bmatrix} = \begin{bmatrix} Cs & CA s & \ldots & CA^{n-1}s \end{bmatrix}, \quad (3.63)$$

then (3.60) automatically holds due to (3.62). By assumption, there exists $s_0 \in \mathbb{R}^n$ such that $\{A, s_0\}$ is controllable or equivalently

$$\text{rk} \left( \begin{bmatrix} s_0 & As_0 & \ldots & A^{n-1}s_0 \end{bmatrix} \right) = n. \quad (3.64)$$

Such $s_0$ is independent of $C$ and $s$ and implies that there always exists $C' \in \mathbb{R}^{1 \times n}$ satisfying (3.63). Thus, Proposition 7 is proven for such matrices $A$.

We now extend the proof of Proposition 7 for general square matrices $A$. Decomposing $A$ in Jordan canonical form

$$A = P \text{ diag}(J_1, \ldots, J_Q) P^{-1}, \quad (3.65)$$

...
where \( J_q \) denotes a Jordan block corresponding to either a real eigenvalue or a pair of complex conjugate eigenvalues, we have, for all \( k \geq 0 \),

\[
CA^k s = \sum_{q=1}^{Q} \bar{C}_q J_q^k \bar{s}_q, \tag{3.66}
\]

where we decompose \( \bar{C} = CP \) and \( \bar{s} = P^{-1}s \) in blocks matching the structure of the Jordan decomposition.

A sufficient condition (but not necessary) for (3.60) to hold is to find, for all \( q \in \{1, \ldots, Q\} \), \( \bar{s}_q^o \) independent of \( C \) and \( s \), and \( \bar{C}'_q \) such that

\[
\bar{C}_q J_q^k \bar{s}_q = \bar{C}'_q J_q^k \bar{s}_q^o, \text{ for all } k \geq 0. \tag{3.67}
\]

Since a Jordan block \( J \) corresponding to a real eigenvalue with, for instance,

\[
s_\circ = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix}
\] \tag{3.68}

or a Jordan block \( J \) corresponding to a pair of complex conjugate eigenvalues with, for instance,

\[
s_\circ = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 \end{bmatrix}
\] \tag{3.69}

satisfies the property that \( \{J, s_\circ\} \) is controllable, such \( \bar{s}_q^o \) and \( \bar{C}'_q \) are found using Proposition 7 for controllable pairs, which concludes the proof.

An immediate consequence of Proposition 7 is that given a matrix \( A \), all the LSSM signals generated by varying \( C \) and \( s \) can be obtained by fixing appropriately the initial state \( s \) and varying \( C \) only.

An analog proposition holds by exchanging the role of \( C \) and \( s \).

**Proposition 8.** Given \( \{C, A, s\} \in \mathbb{R}^{1 \times n} \times \mathbb{R}^{n \times n} \times \mathbb{R}^n \), there exists \( C_\circ \in \mathbb{R}^{1 \times n} \), independent of \( C \) and \( s \), and \( s' \in \mathbb{R}^n \) such that

\[
CA^k s = C_\circ A^k s', \text{ for all } k \geq 0. \tag{3.70}
\]

**Proof of Proposition 8:** The proof follows from Proposition 7 applied to the dual parameterization \( \{s^T, A^T, C^T\} \).

An immediate consequence of Proposition 8 is that given a matrix \( A \), all the LSSM signals generated by varying \( C \) and \( s \) can be obtained by fixing appropriately the observation vector \( C \) and varying \( s \) only.
3.6 A Change of Time-Scale

Given a matrix $A$, Propositions 7 and 8 lead to two interpretations of the LSSM signals obtained by varying $C$ and $s$. Such signals can be interpreted as either a single trajectory generated by a unique $s_0$ projected into the 1-dimensional subspaces spanned by the $C$’s or as all the trajectories generated by all the $s$’s and projected into the same 1-dimensional subspace spanned by $C_0$.

For multi-channel LSSM signals (i.e. when $C$ is a matrix), fixing the initial state (i.e., the trajectory) is more attractive because we can use the same trajectory for each channel but projected into different 1-dimensional subspaces. Handling multiple projections seems more convenient than handling multiple trajectories. This idea is further exploited in Part II.

3.6 A Change of Time-Scale

Given a right-sided LSSM signal $f$ with parameters $\{C, A, s_0\}$ having the structure described in (3.44)–(3.46) (i.e., modified Jordan canonical form), we want to generate and express any time-scaled version of $f$ and find explicit LSSM parameters $\{C^{(\tau)}, A^{(\tau)}, s_0^{(\tau)}\}$. We here assume that $A$ is non-singular (i.e., 0 is not an eigenvalue of $A$).

Let $\tau \in \mathbb{R}^*_+$ be the time scale and $f^{(\tau)}$ be the time-scaled version of $f$ such that for all $k \geq 0$,

$$f^{(\tau)}_k = \sum_{q=1}^{Q} \lambda_q^{k/\tau} \Gamma_q (k/\tau)$$
$$+ \sum_{\ell=1}^{L} \rho^{\ell/\tau} \left( A_\ell (k/\tau) \cos \left( \frac{k\omega_\ell}{\tau} \right) + B_\ell (k/\tau) \sin \left( \frac{k\omega_\ell}{\tau} \right) \right). \quad (3.71)$$

A polynomial $B(k)$ of degree $d-1$ and coefficients $b \in \mathbb{R}^{1 \times d}$ can be written as $B(k) = bk_d$ (cf. (3.13)). Therefore, the time-scaled version of $B(k)$ is simply

$$B(k/\tau) = b D^{(\tau)}_d k_d, \quad (3.72)$$

where

$$D^{(\tau)}_d = \text{diag}(1/\tau^0, 1/\tau^1, \ldots, 1/\tau^{d-1}). \quad (3.73)$$
Using this result and (3.20), we have
\[
\Gamma_q(k/\tau) = C_q T_{\pi_q} D^{(\tau)}_{\pi_q} k_{\pi_q} \quad (3.74)
\]
\[
= C_q T_{\pi_q} D^{(\tau)} T^{-1}_{\pi_q} T k_{\pi_q}. \quad (3.75)
\]
We have analog equalities for \(A_\ell(k/\tau)\) and \(B_\ell(k/\tau)\). Therefore, denoting
\[
C^{(\tau)} = C_q T_{\pi_q} D^{(\tau)} T^{-1}_{\pi_q} \quad (3.76)
\]
\[
C^{(\tau)}_\ell = C_\ell \left( T_{v_\ell} D^{(\tau)} T^{-1}_{v_\ell} \otimes I_2 \right) \quad (3.77)
\]
\[
T = \text{diag} \left( (T_{\pi_q})_{q \in \{1, \ldots, Q\}}, (T_{v_\ell} \otimes I_2)_{\ell \in \{1, \ldots, L\}} \right) \quad (3.78)
\]
\[
D^{(\tau)} = \text{diag} \left( (D^{(\tau)}_{\pi_q})_{q \in \{1, \ldots, Q\}}, (D^{(\tau)}_{v_\ell} \otimes I_2)_{\ell \in \{1, \ldots, L\}} \right), \quad (3.79)
\]
the signal \(f^{(\tau)}\) (the time-scaled version of \(f\)) is generated with the LSSM parameters \(\{C T D^{(\tau)} T^{-1}, A^{(\tau)}, s_0\}\) where \(A^{(\tau)}\) has the same modified Jordan form as \(A\) but where each eigenvalue of \(A\) is raised to the power of \(1/\tau\), i.e., change \(\lambda_q\) with \(\lambda_q^{1/\tau}\), \(\rho_\ell\) with \(\rho_\ell^{1/\tau}\), and \(\omega_\ell\) with \(\omega_\ell/\tau\).

Note that a “pure” polynomial model does not require a modification of its transition matrix \(A\) upon a change of time scale.

### 3.7 Fitting a LSSM Signal

Given measurements \(y_k \in \mathbb{R}, k \in \{0, \ldots, K - 1\}\), we want to fit a right-sided LSSM signal of a given order \(n \in \mathbb{N}\) by minimizing the squared error
\[
J(\theta_0) = \sum_{k=0}^{K-1} (y_k - C A^k s)^2, \quad (3.80)
\]
with respect to the LSSM parameters \(\theta_0 = \{C, A, s\}\). Since there is an ambiguity in the LSSM parameters \(\theta_0\), we often use a specific parameterization, such as the observable or controllable or Jordan canonical forms to ensure uniqueness of the parameters.

Standard algorithms for minimizing \(J(\theta_0)\), such as gradient descent, can be used with some effort. As an alternative, we propose a statistical method. We define the likelihood function
\[
p(y|\theta_0, \sigma^2_Z) = \frac{1}{\sqrt{(2\pi\sigma^2_Z)^K}} \exp \left( -\frac{J(\theta_0)}{2\sigma^2_Z} \right). \quad (3.81)
\]
Minimizing $J(\theta_0)$ is equivalent to maximizing the likelihood for a fixed $\sigma_Z^2$ or also maximizing the likelihood with respect to both $\theta_0$ and $\sigma_Z^2$ since

$$\hat{\sigma}_Z^2 = \arg\max_{\sigma_Z^2} p(y|\theta_0, \sigma_Z^2) = \frac{J(\theta_0)}{K}$$  \hspace{1cm} (3.82)

$$\max_{\sigma_Z^2} \ln p(y|\theta_0, \sigma_Z^2) = -\frac{K}{2} - \frac{K}{2} \ln \left(2\pi \frac{J(\theta_0)}{K}\right).$$  \hspace{1cm} (3.83)

The likelihood function can be trivially written as

$$p(y|\theta_0, \sigma_Z^2) = \int \delta(x_0 - s) \prod_{k=1}^{K-1} \delta(x_k - Ax_{k-1}) \prod_{k=0}^{K-1} \mathcal{N}(y_k : Cx_k, \sigma_Z^2) \, dx,$$  \hspace{1cm} (3.84)

where $x = (x_0, \ldots, x_{K-1})$. In this expression, the joint density $p(y, x|\theta_0, \sigma_Z^2)$ trivially factorizes. Due to the hard constraints (delta-Dirac distributions) linking the hidden variables $x_0, \ldots, x_{K-1}$, many variational estimation algorithms would get stuck at the initial points. In order to circumvent this problem, we approximate the delta-Dirac distribution with a zero-mean Gaussian density of covariance matrix $V_E \in \mathbb{S}_+^*$. The new likelihood becomes

$$p(y|\theta) = \int \mathcal{N}(x_0 : s, V_E) \prod_{k=1}^{K-1} \mathcal{N}(x_k : Ax_{k-1}, V_E)$$

$$\cdot \prod_{k=0}^{K-1} \mathcal{N}(y_k : Cx_k, \sigma_Z^2) \, dx,$$  \hspace{1cm} (3.85)

with $\theta = \theta_0 \cup \{\sigma_Z^2, V_E\}$. Note that both $\sigma_Z^2$ and $V_E$ regularize the estimation of $\theta_0$. Indeed, the likelihood expresses as

$$p(y|\theta) = \frac{1}{\sqrt{(2\pi)^{K(n+1)} \sigma_Z^{2K} |V_E|^K}} \int \exp \left(-\frac{1}{2} J(x, \theta) \right) \, dx,$$  \hspace{1cm} (3.86)

with the cost function

$$J(x, \theta) = (x_0 - s)^T V_E^{-1} (x_0 - s) + \sum_{k=0}^{K-1} \frac{(y_k - Cx_k)^2}{\sigma_Z^2}$$

$$+ \sum_{k=1}^{K-1} (x_k - Ax_{k-1})^T V_E^{-1} (x_k - Ax_{k-1}).$$  \hspace{1cm} (3.87)
Since $J(x, \theta)$ is a quadratic form in $x$, we further have

$$\int \exp \left( -\frac{1}{2} J(x, \theta) \right) \, dx = \exp \left( -\frac{1}{2} \min_x J(x, \theta) \right) \sqrt{\frac{(2\pi)^K n}{|W_X|}}, \quad (3.88)$$

where $W_X$ is the precision matrix associated to $X$, which is a block-tridiagonal square matrix of dimension $K \cdot n$ and depending on $\theta$.

It follows that

$$p(y|\theta) = \frac{1}{\sqrt{(2\pi)^K \sigma^2_Z |V_E|^K |W_X|}} \exp \left( -\frac{1}{2} \min_x J(x, \theta) \right). \quad (3.89)$$

By introducing state noise (via the covariance matrix $V_E$), we actually allow deviations of the state trajectory $x_1, \ldots, x_K$ from the autonomous trajectory $\hat{x}_k = A^k s$ specifically, for $V_E \neq 0$, the MMSE/MAP estimate of the state trajectory $X_1, \ldots, X_K$ no longer necessarily lies on an autonomous trajectory of the form $\hat{x}_k = A^k s$.

In order to maximize the likelihood $p(y|\theta)$, we now provide an expectation maximization (EM) algorithm [24]. Using the integral form (3.85), the EM algorithm iteratively updates $\theta$ with

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E} [\ln p(y|x, \theta)p(x|\theta)], \quad (3.90)$$

where the expectation is computed with respect to $p(x|y, \hat{\theta}^{\text{old}})$. Expanding the expectation term, we have

$$-2 \mathbb{E} [\ln p(y|x, \theta)p(x|\theta)]$$

$$= K \ln(2\pi \sigma^2_Z) + \frac{1}{\sigma^2_Z} \sum_{k=0}^{K-1} \mathbb{E} [(y_k - CX_k)^2]$$

$$+ K \ln((2\pi)^n|V_E|) + \mathbb{E} [(X_0 - s)^T V_E^{-1} (X_0 - s)]$$

$$+ \sum_{k=1}^{K-1} \mathbb{E} [(X_k - AX_{k-1})^T V_E^{-1} (X_k - AX_{k-1})] \quad (3.91)$$

$$= K \ln(2\pi \sigma^2_Z) + \frac{1}{\sigma^2_Z} (\kappa - 2C \mathcal{E}_{X,y} + C \mathcal{E}_{X,X} C^T) + Kn \ln(2\pi)$$

$$+ K \ln |V_E| + \text{tr} \left( V^{-1}_E \left( ss^T - 2s \mathbb{E} [X_0]^T + \mathbb{E} [X_0 X_0^T] \right) \right)$$

$$+ \text{tr} \left( V^{-1}_E \left( A \mathcal{E}_{X,X} A^T - 2A \mathcal{E}_{X,X} + \mathcal{E}'_{X,X} \right) \right), \quad (3.92)$$
with
\[
\kappa = \sum_{k=0}^{K-1} y_k^2
\]  
(3.93)
\[
\mathcal{E}_{X,y} = \sum_{k=0}^{K-1} \mathbb{E}[X_k]y_k
\]  
(3.94)
\[
\mathcal{E}_{X,X} = \sum_{k=0}^{K-1} \mathbb{E}[X_kX_k^T]
\]  
(3.95)
\[
\mathcal{E}'_{X,X} = \sum_{k=1}^{K-1} \mathbb{E}[X_kX_k^T]
\]  
(3.96)
\[
\mathcal{E}_{X-,X} = \sum_{k=1}^{K-1} \mathbb{E}[X_{k-1}X_k^T]
\]  
(3.97)
\[
\mathcal{E}_{X-,X-} = \sum_{k=1}^{K-1} \mathbb{E}[X_{k-1}X_{k-1}^T].
\]  
(3.98)

Thus, the maximization in the EM step \[3.90\] splits for \((C, \sigma_Z^2)\) on the one hand and \((A, s, V_E)\) on the other hand. For updating \((C, \sigma_Z^2)\), we obtain
\[
\hat{C} = \arg\min_C C\mathcal{E}_{X,X}C^T - 2C\mathcal{E}_{X,y}
\]  
(3.99)
\[
\hat{\sigma}_Z^2 = \frac{1}{K}\left(\kappa - 2\hat{C}\mathcal{E}_{X,y} + \hat{C}\mathcal{E}_{X,X}\hat{C}^T\right).
\]  
(3.100)

Unless there are specific needs, the covariance matrix \(V_E\) can be assumed to be of the form \(V_E = \sigma_E^2 I_n\) since we anyway hope that \(V_E\) goes to zero. Then, the EM rules for \((A, s, V_E)\) become
\[
\hat{A} = \arg\min_A \text{tr} \left(A\mathcal{E}_{X-,X-}A^T - 2A\mathcal{E}_{X-,X}\right)
\]  
(3.101)
\[
\hat{s} = \mathbb{E}[X_0]
\]  
(3.102)
\[
\hat{\sigma}_E^2 = \frac{1}{nK} \left(\text{tr} \left(\hat{A}\mathcal{E}_{X-,X-}\hat{A}^T - 2\hat{A}\mathcal{E}_{X-,X} + \mathcal{E}_{X,X}\right) - \|\mathbb{E}[X_0]\|^2\right).
\]  
(3.103)

Note that \(C\) does not usually need to be estimated and can be fixed when \(s\) is estimated, as a consequence of the over-parametrization explained in Sec. 3.5. The minimization problem in \[3.101\] consists of minimizing a general quadratic form and closed-form solutions are available even when
A is in observable or controllable or Jordan canonical form, as derived in Appendix D.

In addition, all expectation quantities and the likelihood \( p(y|\theta) \) can be computed with Gaussian message passing, as described in Chapter 4.

### 3.8 Inner Product With LSSM Signals

Given a scalar two-sided LSSM signal \( f \) with parameters \( \{C, A, s\} \) and any discrete-time signal \( y = (y_1, \ldots, y_K) \in \mathbb{R}^K \) of duration \( K \in \mathbb{N} \), we define the quantity

\[
\xi_k(y, A, s) = \begin{bmatrix}
\xi_k(y, A_p, s_p) \\
\xi_k(y, A_f, s_f)
\end{bmatrix} \in \mathbb{R}^n,
\tag{3.104}
\]

with \( n = n_p + n_f \) and

\[
\xi_k(y, A_p, s_p) = \sum_{i=1}^{k} A_p^{k-i} s_p y_i \in \mathbb{R}^{n_p}
\tag{3.105}
\]

\[
\xi_k(y, A_f, s_f) = \sum_{i=k+1}^{K} A_f^{i-k} s_f y_i \in \mathbb{R}^{n_f},
\tag{3.106}
\]

for \( k \in \{1, \ldots, K\} \) and \( \xi_k(y, A, s) = 0 \), otherwise. Note that \( \xi_k(y, A, s) \) is a linear function of \( y \) and can be interpreted as the output of \( n \) linear filters.

The quantity (3.104) is efficiently computed for all \( k \in \{1, \ldots, K\} \) using the forward recursion

\[
\xi_k(y, A_p, s_p) = A_p \xi_{k-1}(y, A_p, s_p) + s_p y_k,
\tag{3.107}
\]

initialized with \( \xi_0(y, A_p, s_p) = 0 \) and the backward recursion

\[
\xi_k(y, A_f, s_f) = A_f (\xi_{k+1}(y, A_f, s_f) + s_f y_{k+1}),
\tag{3.108}
\]

initialized with \( \xi_K(y, A_f, s_f) = 0 \). When both \( A_p \) and \( A_f \) are stable (i.e., all their eigenvalues are inside the open unit disc in the complex plane), both recursions (3.107) and (3.108) are numerically stable (see the analysis in Sec. 6.2.4).
Proposition 9 (Inner Product with a LSSM signal). The inner product between \( y \) and a LSSM signal \( f(C) \), for any \( C \), as in (3.1) shifted by a time \( k \) and denoted by \( f_{\bullet-k}(C) \) is

\[
\langle y, f_{\bullet-k}(C) \rangle = C\xi_k(y, A, s).
\] (3.109)

Recall that \( C = [C] = [C_p \quad C_f] \).

Proof of Proposition 9 \( \exists \). This proposition follows from

\[
\langle y, f_{\bullet-k}(C) \rangle = \sum_{i=1}^{K} y_i f_{i-k}(C)
\] (3.110)

\[
= \sum_{i=1}^{k} C_p A_p^{k-i} s_p y_i + \sum_{i=k+1}^{K} C_f A_f^{i-k} s_f y_i
\] (3.111)

\[
= C_p \xi_k(y, A_p, s_p) + C_f \xi_k(y, A_f, s_f)
\] (3.112)

\[
= C\xi_k(y, A, s).
\] (3.113)

Proposition 9 has several important consequences. First, computing the correlation function between a signal of length \( K \) and a LSSM signal \( f(C) \) has a complexity of \( O(Kn^2) \), with \( n = \max(n_p, n_f) \). Secondly, the inner product between \( y \) and \( f_{\bullet-k}(C) \) can be expressed as a standard inner product in \( \mathbb{R}^n \) between \( C \) and \( \xi_k(y, A, s) \). Finally, since \( \xi_k(y, A, s) \) is independent of \( C \), the computational effort to obtain the inner product between \( y \) and \( f_{\bullet-k}(C) \) for any \( C \), is of \( O(n) \) only, after having computed \( \xi_k(y, A, s) \).
Chapter 4

Computations in Linear State Space Models

Let $M \in \mathbb{N}$. Let $y_1, \ldots, y_K \in \mathbb{R}^M$ be $M$-channel observations of duration $K \in \mathbb{N}$, which we want to explain with a LSSM as in (2.1) of order $n \in \mathbb{N}$ and given by

$$
\begin{cases}
X_k = AX_{k-1} + B_k U_k + E_k \\
y_k = CX_k + Z_k,
\end{cases} \quad (4.1)
$$

with (mutually independent) Gaussian random vector variables

$$
X_0 \sim \mathcal{N}(m_0, V_0) \quad (4.2)
$$

$$
Z_k \overset{iid}{\sim} \mathcal{N}(0, \sigma_Z^2 I_M) \quad (4.3)
$$

$$
U_k \sim \mathcal{N}(0, \Sigma_U) \quad (4.4)
$$

$$
E_k \overset{iid}{\sim} \mathcal{N}(0, V_E), \quad (4.5)
$$

and parameters $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{M \times n}$, $B_k \in \mathbb{R}^{n \times J}$ (with $J \in \mathbb{N}$), $m_0 \in \mathbb{R}^n$, $\sigma_Z^2 > 0$, and symmetric positive (semi-)definite matrices $V_0 \in \mathbb{R}^{n \times n}$, $\Sigma_U \in \mathbb{R}^{J \times J}$, and $V_E \in \mathbb{R}^{n \times n}$.

Given this LSSM, we want to efficiently compute the joint posterior density of $X_{k-1}, X_k,$ and $U_k$ for any $k \in \{1, \ldots, K\}$. Furthermore, given all the (LSSM and statistical) parameters (denoted by $\theta$), we wish to compute the likelihood $p(y_1, \ldots, y_K | \theta)$. 

37
4.1 Gaussian Message Passing

4.1.1 Generality

Since all random vectors (4.2)–(4.5) are Gaussian and all random vectors are linked via linear constraints (4.1), it results that, given \( \theta \), the random vectors \( X = (X_1, \ldots, X_K) \), \( U = (U_1, \ldots, U_K) \), \( Z = (Z_1, \ldots, Z_K) \), and \( E = (E_1, \ldots, E_K) \) are jointly Gaussian. Thus, all posterior densities are characterized by its mean and covariance matrix, which can be efficiently computed using a Gaussian message passing algorithm in the factor graph [45] of Fig. 4.1.

There are several ways of performing Gaussian message passing [11,44,45]. Each of them use a different parametrization of a Gaussian density (e.g., mean/variance, weighted mean/precision matrix, dual mean/dual precision) and comes with its own advantages and drawbacks. The main issue of such algorithms is numerical instability, which typically arises when

- some of the parameters, such as (co)variance inputs, are too small or too large,

- the LSSM order and/or the number of channels are too big,
• matrix inversions are required,
• the matrix $A$ is not stable.

In most of our applications, we will indeed experience small (or exact zero) input covariances, a relatively high number of channels, and huge state space orders. For that reason, we will often use the modified Bryson-Frazier smoother (cf. [13, 44]) as described in the next section, since matrix inversions are avoided and numerical stability is preserved when input variances are small.

### 4.1.2 Modified Bryson-Frazier Smoother

We here recall the modified Bryson-Frazier smoother algorithm [7, 13, 44]. Compared to the standard message passing algorithm that propagates the forward mean $\rightarrow m_{X_k}$ and covariance $\rightarrow V_{X_k}$ and (possibly simultaneously) the backward mean $\leftarrow m_{X_k}$ and covariance $\leftarrow V_{X_k}$, the modified Bryson-Frazier smoother first consists of a forward sweep to propagate the forward mean $\rightarrow m_{X_k}$ and covariance $\rightarrow V_{X_k}$ while storing extra quantities ($G_k$ and $F_k$) and concludes with a backward sweep that propagates the dual mean $\sim \xi_{X_k}$ and dual precision matrix $\sim W_{X_k}$. These quantities satisfy the relation

\[
\sim \xi_{X_k} = \sim W_{X_k} (\rightarrow m_{X_k} - \leftarrow m_{X_k}) \tag{4.6}
\]
\[
\sim W_{X_k} = \left( \rightarrow V_{X_k} + \leftarrow V_{X_k} \right)^{-1}. \tag{4.7}
\]

More specifically, the smoother proceeds in two sweeps. The forward pass for $k = 1, \ldots, K$,

\[
\rightarrow m_{X_k} = A(\rightarrow m_{X_{k-1}} + \rightarrow V_{X_{k-1}} C^T G_{k-1}(y_{k-1} - C\rightarrow m_{X_{k-1}})) \tag{4.8}
\]
\[
\rightarrow V_{X_k} = B_k \Sigma U_k B_k^T + V_E + A F_{k-1} \rightarrow V_{X_{k-1}} A^T \tag{4.9}
\]
\[
G_k = (\sigma_Z^2 I_M + C\rightarrow V_{X_k} C^T)^{-1} \tag{4.10}
\]
\[
F_k = I_n - \rightarrow V_{X_k} C^T G_k C, \tag{4.11}
\]

with initialization $\rightarrow m_{X_0} = m_0$, $\rightarrow V_{X_0} = V_0$, and $y_0 = 0$.

Then, the backward pass for $k = K, \ldots, 1$,

\[
\sim \xi_{X_k} = F_k^T A^T \sim \xi_{X_{k+1}} - C^T G_k (y_k - C\rightarrow m_{X_k}) \tag{4.12}
\]
\[
\sim W_{X_k} = F_k^T A^T \sim W_{X_{k+1}} A F_k + C^T G_k C, \tag{4.13}
\]
initialized with $\tilde{W}_{X_{K+1}} = 0$ and $\tilde{\xi}_{X_{K+1}} = 0$. These initializations reflect the fact that no prior information (from the right part of the graph in Fig. 4.1) on the state variable $X_K'$ is available, which is accordingly enforced by choosing an infinite backward covariance matrix $\tilde{V}_{X_K'}$.

Note that step (4.10) does not actually require a matrix inversion and can be computed with a loop of size $M$ involving matrix multiplications and scalar inversions only. Indeed, the equality node in Fig. 4.1 involving vector observations can be split to get scalar observations since the noise term $Z_k$ is uncorrelated, leading to Fig. 4.2. Thus, $\tilde{V}_{X_K'}$ can be computed iteratively as follows. Initialize $\tilde{V}_{X_K^{(o)}} = V_{X_k}$ and then, for $m = 1, \ldots, M$, compute

$$
\tilde{V}_{X_k^{(m)}} = \tilde{V}_{X_k^{(m-1)}} - \frac{\tilde{V}_{X_k^{(m-1)}} C_m C_m^T \tilde{V}_{X_k^{(m-1)}}}{\sigma_Z^2 + C_m \tilde{V}_{X_k^{(m-1)}} C_m^T},
$$

where $C_m$ is the $m$th row of $C$. Finally, we have

$$
\tilde{V}_{X_K'} = \tilde{V}_{X_k^{(M)}} = \tilde{V}_{X_k^{(M)}}
$$

Actually, even if the noise $Z_k$ is correlated with a covariance matrix $V_Z$, the same trick can be applied by pre-whitening the observations: replace $V_Z$, $C$, and $y_k$ by $I_M$, $S_Z^{-1}C$, and $S_Z^{-1}y_k$, respectively, with $S_Z$ such that $V_Z = S_Z S_Z^T$ (e.g., $S_Z$ obtained by a Cholesky decomposition of $V_Z$).
4.1.3 Improving Numerical Stability

In some cases, and especially when using two-sided models, the proposed message passing can lead to very poor condition numbers for the matrices $\tilde{V}_{X_k}$, $F_k$, and $\tilde{W}_{X_k}$. The reason for this matter is that some eigenvalues of $A$ have a module greater or equal than one while other eigenvalues of $A$ have a module strictly smaller than one. Then, the mixing effects in (4.9) and (4.13) tend to increase the condition numbers of $\tilde{V}_{X_k}$, $F_k$, and $\tilde{W}_{X_k}$ as the iterations go, which creates numerical stability issues.

Using the ideas proposed in [11], we seek a transform matrix $T \in \mathbb{R}^{n \times n}$ (invertible) such that the new state variables

$$\tilde{X}_k = TX_k,$$ 

(4.17)

generates better-conditioned matrices

$$\tilde{V}_{\tilde{X}_k} = T\tilde{V}_{X_k}T^\top$$

(4.18)

$$\tilde{W}_{\tilde{X}_k} = (T^{-1})^\top\tilde{W}_{X_k}T^{-1}$$

(4.19)

$$\tilde{F}_k = T\tilde{F}_kT^{-1}$$

(4.20)

for $k \in \{1, \ldots, K\}$, with the transformed LSSM parameters

$$\tilde{C} = CT^{-1}$$

(4.21)

$$\tilde{A} = TAT^{-1}$$

(4.22)

$$\tilde{B}_k = TB_k$$

(4.23)

$$\tilde{V}_E = TV_ET^\top.$$ 

(4.24)

Note that $G_k$ is invariant to a change of state space parametrization. In addition, since both $\tilde{V}_{X_k}$ and $\tilde{W}_{X_k}$ are symmetric positive semi-definite matrices, it results that the product $\tilde{V}_{X_k}\tilde{W}_{X_k}$ is diagonalizable [33] and, as a consequence of

$$\tilde{V}_{\tilde{X}_k}\tilde{W}_{\tilde{X}_k} = T\tilde{V}_{X_k}\tilde{W}_{X_k}T^{-1},$$

(4.25)

the eigenvalues of $\tilde{V}_{X_k}\tilde{W}_{X_k}$ are also invariant to a change of state space parametrization.

Neglecting for a moment the input covariances $\Sigma_{U_k}$, we define the steady-state matrix $\tilde{V}_{ss}$ as a steady solution of the forward covariance recursion (4.9), i.e.,

$$\tilde{V}_{ss} = V_E + A(\tilde{V}_{ss} - \tilde{V}_{ss}C^\top(\sigma_2^2 I_M + C\tilde{V}_{ss}C^\top)^{-1}C\tilde{V}_{ss})A^\top,$$ 

(4.26)
which is a discrete algebraic Riccati equation (DARE). A solution to the DARE might not exist but we here assume that there is one and that it is symmetric positive definite. We refer to [35, Appendix E] for further details. Given $V_{ss}$, we also define the steady-state matrices

$$G_{ss} = (\sigma_Z^2 I_M + C V_{ss} C^T)^{-1}$$

$$F_{ss} = I_n - V_{ss} C^T G_{ss} C.$$  \hspace{1cm} (4.27) \hspace{1cm} (4.28)

Similarly, the steady-state matrix $W_{ss}$ of the dual precision recursion (4.13) is solution of

$$\tilde{W}_{ss} = F_{ss}^T A^T \tilde{W}_{ss} A F_{ss} + C^T G_{ss} C,$$  \hspace{1cm} (4.29)

which is a discrete Lyapunov equation. Again a solution might not exist [35, Appendix D] but we here assume that there is one and that it is symmetric positive definite.

All the time-dependent matrices $V_{X_k}, F_k$, and $W_{X_k}$ are somehow close to their steady-state matrices (especially when $\Sigma_{U_k}$ is often zero). Thus, having well-conditioned steady-state matrices help to make recursions more stable numerically.

To this end, since both $V_{ss}$ and $W_{ss}$ are symmetric positive definite, there exists a transform $P$ which simultaneously diagonalizes $V_{ss}$ and $W_{ss}$ [33], i.e.,

$$P V_{ss} P^T = \Lambda$$

$$(P^{-1})^T W_{ss} P^{-1} = \Gamma,$$  \hspace{1cm} (4.30) \hspace{1cm} (4.31)

where both $\Lambda$ and $\Gamma$ are diagonal matrices with strictly positive diagonal elements. As an immediate consequence, since $V_{ss} W_{ss}$ is diagonalizable, $\Lambda \Gamma$ is a diagonal matrix containing in its diagonal the eigenvalues of $V_{ss} W_{ss}$, which are invariant to a change of state space parametrization.

Then, by inspecting the diagonal elements of $\Lambda \Gamma$, we choose a diagonal matrix $D$ such that both $D \Lambda D^T$ and $(D^{-1})^T \Gamma D^{-1}$ are well-conditioned and thus choosing

$$T = DP,$$  \hspace{1cm} (4.32)

as a state space transform will improve the numerical stability.

In particular, it is proposed in [11] to choose

$$D = (\Lambda^{-1} \Gamma)^{\frac{1}{2}},$$  \hspace{1cm} (4.33)
such that the transformed matrices are evenly conditioned since
\[ T\tilde{V}_{ss}T^T = (T^{-1})^T\tilde{W}_{ss}T^{-1} = (\Lambda\Gamma)^{\frac{1}{2}}. \] (4.34)

The main drawbacks of this stabilization method are that solutions to the equations (4.26) and (4.29) not only need to exist but also need to be computed in a numerically robust way.

### 4.1.4 Processing LSSM Models defined Forward and Backward in Time

In this section, we focus on applying the modified Bryson-Frazier smoother to the LSSMs defined backward in time (2.25) and forward/backward in time (2.26). When confronted with such models, the initialization of messages need to be carefully addressed. In addition, numerical instabilities can arise and require the use of a stabilization method, such as the one described in Sec. 4.1.3.

#### LSSM Backward in Time

For a LSSM defined backward in time (2.25), the initial backward messages for \( X_K \) are set according to the specified prior distribution on \( X_K \) while the initial forward messages on \( X_0 \) are set such that no information is carried (i.e., an infinite forward covariance matrix \( \tilde{V}_{X_0} \rightarrow +\infty \)).

A natural adaptation of the modified Bryson-Frazier smoother to a LSSM defined backward in time consists of reversing the recursions: first perform a backward sweep to compute the backward covariance matrices (starting with the specified values of \( \tilde{m}_{X_K} \) and \( \tilde{V}_{X_K} \)) and then perform a forward sweep to compute the dual precision matrices (starting with \( \tilde{W}_{X_0} = 0 \)). The recursions do not exactly coincide with the ones given in Sec. 4.1.2 but can easily be derived from the tables in [44].

Alternatively, rewriting (2.25) by simply reversing the time direction, we obtain
\[
\begin{cases}
X_k = A^{-1}X_{k-1} - B_kU_k - E_k \\
y_k = CX_k + Z_k,
\end{cases}
\] (4.35)
assuming that \( A \) is invertible. Even if this reversed model is not fully equivalent to the one in (2.25) with respect to all the statistical assumptions, the posterior distributions computed with the standard modified Bryson-Frazier smoother for the model (4.35) with forward initialization...
\[ \tilde{V}_{X_0} \rightarrow +\infty \] and backward initializations at \( k = K \) using (4.6) and (4.7) coincide with the posterior distributions of the model (2.25). However, since the forward initialization cannot be set in practice, the posterior messages are approximations of the true posterior messages. If the system is well-behaved, the effects of an incorrect initialization decrease as the time index increases.

**LSSM Forward/Backward in Time**

For a LSSM defined forward/backward in time (2.26), the messages \( \tilde{m}_{X_0} \), \( \tilde{V}_{X_0} \), \( \tilde{m}_{X_{K+1}} \), and \( \tilde{V}_{X_{K+1}} \) are specified by the model while the covariance matrices \( \tilde{V}_{X_0} \) and \( \tilde{V}_{X_{K+1}} \) are assumed to be infinite.

It is not trivial to exactly handle these initializations in a message passing algorithm. However, obtaining a good approximation of the posterior messages is possible by rewriting (2.26) with a unique time direction, that is to say,

\[
\begin{align*}
X_k &= \begin{bmatrix} A_f & 0 \\ 0 & A_p^{-1} \end{bmatrix} X_{k-1} + \begin{bmatrix} B_f^t \\ -B_p^t \end{bmatrix} U_k + \begin{bmatrix} E_f^t \\ -E_p^t \end{bmatrix} \\
y_k &= \begin{bmatrix} C_f \\ C_p \end{bmatrix} X_k + Z_k,
\end{align*}
\] (4.36)

assuming that \( A_p \) is invertible and with

\[
X_k = \begin{bmatrix} X_{k}^f \\ X_{k}^p \end{bmatrix}.
\] (4.37)

Then, the standard modified Bryson-Frazier smoother for the model (4.36) can be performed while including the initialization of the messages as described above. This approach often leads to numerical instabilities that become increasingly harder to handle as the system order grows.

### 4.2 From Messages to Expectations

The posterior and joint posterior of \( X_{k-1}, U_k, \) and \( X_k \) are all Gaussian and thus characterized by their mean and covariance matrix. When using an EM algorithm, the expectations of these variables are of great importance. We here provide the links between these expectations and the quantities obtained from the modified Bryson-Frazier smoother.
The posterior density of $X_k$ is given by
\[
m_{X_k} = \bar{m}_{X_k} - \bar{V}_{X_k} \xi_{X_k}
\]
\[
V_{X_k} = \bar{V}_{X_k} (I_n - \bar{W}_{X_k} \bar{V}_{X_k}).
\]

The posterior density of $U_k$ is parameterized by
\[
m_{U_k} = -\Sigma_{U_k} B_k^T \xi_{X_k}
\]
\[
V_{U_k} = \Sigma_{U_k} - \Sigma_{U_k} B_k^T \bar{W}_{X_k} B_k \Sigma_{U_k}
\]
\[
\mathbb{E}[U_k U_k^T] = \Sigma_{U_k} + \Sigma_{U_k} B_k^T (\xi_{X_k} \xi_{X_k}^T - \bar{W}_{X_k}) B_k \Sigma_{U_k}.
\]

The joint posterior of $X_{k-1}$ and $X_k$ is characterized by the cross-covariance matrix
\[
V_{X_{k-1}, X_k^T} = F_{k-1} \bar{V}_{X_{k-1}} A^T (I_n - \bar{W}_{X_k} \bar{V}_{X_k})
\]
and the expected cross-correlation is
\[
\mathbb{E}[X_{k-1} X_k^T] = V_{X_{k-1}, X_k^T} + m_{X_{k-1}} m_{X_k}^T.
\]

The joint posterior density of $X_{k-1}$ and $U_k$ is characterized by the cross-covariance matrix
\[
V_{X_{k-1}, U_k} = -F_{k-1} \bar{V}_{X_{k-1}} A^T \bar{W}_{X_k} B_k \Sigma_{U_k},
\]
and the expected cross-correlation is
\[
\mathbb{E}[X_{k-1} U_k^T] = -(F_{k-1} \bar{V}_{X_{k-1}} A^T \bar{W}_{X_k} + m_{X_{k-1}} \xi_{X_k}^T) B_k \Sigma_{U_k}.
\]

The joint posterior density of $X_k$ and $U_k$ is characterized by the cross-covariance matrix
\[
V_{X_k, U_k} = (I_n - \bar{V}_{X_k} \bar{W}_{X_k}) B_k \Sigma_{U_k},
\]
and the expected cross-correlation is
\[
\mathbb{E}[X_k U_k^T] = (I_n - \bar{V}_{X_k} \bar{W}_{X_k} - m_{X_k} \xi_{X_k}^T) B_k \Sigma_{U_k}.
\]

Note that all those expectation computations do not require any matrix inversions. While most of these formulas are easily derived using [44], the formulas for the cross-covariance matrices are derived in Appendix C.
4.3 Likelihood Computation

Given $\theta$, the log-likelihood $L_K = \ln p(y_1, \ldots, y_K | \theta)$ can be efficiently computed using Proposition 10.

**Proposition 10** (Likelihood computation). For a fixed $\theta$, the log-likelihood

$$L_k(\theta) = \ln p(y_1, \ldots, y_k | \theta)$$

(4.49)

can be computed iteratively using the recursion

$$L_k = L_{k-1} - \frac{M}{2} \ln(2\pi) + \frac{1}{2} \ln |G_k| - \frac{1}{2} (y_k - C\vec{m}_X^k)^T G_k (y_k - C\vec{m}_X^k),$$

(4.50)

with $L_0 = 0$.

**Proof of Proposition 10**: This formula follows from

$$p(y_1, \ldots, y_k | \theta) = p(y_k | y_1, \ldots, y_{k-1}, \theta) p(y_1, \ldots, y_{k-1} | \theta).$$

(4.51)

Furthermore, we have

$$p(y_k | y_1, \ldots, y_{k-1}, \theta)$$

$$= \int p(y_k | y_1, \ldots, y_{k-1}, x_k, \theta) p(x_k | y_1, \ldots, y_{k-1}, \theta) \, dx_k$$

(4.52)

$$= \int \mathcal{N}(y_k : Cx_k, \sigma^2 Z I_M) \mathcal{N}(x_k : \vec{m}_X^k, \vec{V}_X^k) \, dx_k$$

(4.53)

$$= \mathcal{N}(y_k : C\vec{m}_X^k, \sigma^2 Z I_M + C\vec{V}_X^k C^T)$$

(4.54)

$$= \mathcal{N}(y_k : C\vec{m}_X^k, G_k^{-1}),$$

(4.55)

(using (4.10)) from which we deduce (4.50).
Part II

Detection and Estimation in Multi-Channel Signals With Autonomous Linear State Space Models
“Avant donc que d’écrire, apprenez à penser.
Ce que l’on conçoit bien s’énonce clairement,
Et les mots pour le dire arrivent aisément.

Hâtez-vous lentement, et sans perdre courage,
Vingt fois sur le métier remettez votre ouvrage,
Polissez-le sans cesse, et le repolissez,
Ajoutez quelquefois, et souvent effacez.”

“Even before writing, learn how to think.
What is conceived well is expressed clearly,
And the words to say it come easily.

Hasten slowly, and without losing heart,
Put your work twenty times upon the anvil,
Keep refine it, and refine it again,
Add from time to time, and often erase.”

Extraits, L’Art poétique, Nicolas Boileau.
Chapter 5

Events in Multi-Channel Signals: Definition, Examples, and Problem Statement

When processing single-channel or multi-channel discrete-time signals, we are often confronted with the problem of detecting events and estimating parameters of such events. We develop a general approach for such detection and estimation problem based on the use of autonomous LSSMs. The proposed approach was outlined in [62] and [12], see also [25,61], but is here developed much further. Other approaches for event detection and estimation include [39,59].

An event is defined as a change between a “past” model and a “future” model at a given formal location \( k \). Conditioned on an event occurring at time index \( k \), the pivotal idea of the proposed approach is to describe the target signal \( \tilde{y}_1, \tilde{y}_2, \ldots \in \mathbb{R}^M \) by two autonomous LSSMs as follows. The first LSSM of order \( n_f \in \mathbb{N} \) (with state \( x_i \in \mathbb{R}^{n_f} \)) describes the target signal for \( i > k \):

\[
\begin{align*}
x_i &= A_f x_{i-1} \\
\tilde{y}_i &= C_f x_i,
\end{align*}
\]

with \( A_f \in \mathbb{R}^{n_f \times n_f} \) and \( C_f \in \mathbb{R}^{M \times n_f} \). The second state space model of order \( n_p \in \mathbb{N} \) (with state \( x'_i \in \mathbb{R}^{n_p} \)), generated backward in time,
describes the target signal for \( i \leq k \):

\[
x'_{i-1} = A_p x'_i \\
\bar{y}_i = C_p x'_i,
\]

with \( A_p \in \mathbb{R}^{n_p \times n_p} \) and \( C_p \in \mathbb{R}^{M \times n_p} \). (The subscripts “p” and “f” stand for “past” and “future”, respectively.) Note that these state space models have no input (hence “autonomous”). In consequence, the states at the formal location \( k \), i.e., \( x'_k = s_p \) and \( x_k = s_f \), fully determine \( x'_i \), \( x_i \), and \( \bar{y}_i \) at all times \( i \):

\[
\bar{y}_i = \begin{cases} 
C_p A_{k-i-p} s_p & \text{for } i \leq k \\
C_f A_{i-k-f} s_f & \text{for } i > k.
\end{cases}
\]

Thus, the target signal \( \bar{y}_i, i \in \mathbb{Z} \), is a multi-channel LSSM signal shifted by a time \( k \). With this definition, an event can be interpreted in various ways, as illustrated in Fig. 5.1. Indeed, the effect of an event can be seen as, for instance,

1. a change of slope in a piecewise linear signal,
2. a sharp onset of a decaying sinusoid,
3. a two-sided fastly-decaying sinusoid,
4. a two-sided fastly-decaying sinusoid on top of a wandering baseline.

All those examples are single-channel cases but generalize to the multi-channel case.

In the following, we assume that the transition matrices \( A_p \) and \( A_f \) are fixed and that the set of allowed events are characterized by varying \( C_p, s_p, C_f, \) and \( s_f \). However, as a consequence of Sec. 3.5, it is enough to vary \( C_p \) and \( C_f \) only while leaving \( s_p \) and \( s_f \) fixed (in a suitable manner). Thus, estimating the parameters of an event boils down to estimating the observation matrix

\[
C \triangleq \begin{bmatrix} C_p & C_f \end{bmatrix} \in \mathcal{C},
\]

where \( \mathcal{C} \) is a given subset of \( \mathbb{R}^{M \times n} \) (with \( n \triangleq n_p + n_f \)) that encodes the admissible events. For instance, referring to Fig. 5.1 in the examples “change of slope” and “two-sided fastly-decaying sinusoid” (with or without baseline), \( \mathcal{C} \) can encode that a discontinuity is not allowed between the “past” and “future” models. Such continuity constraint expresses as

\[
C_p s_p = C_f s_f,
\]
Figure 5.1 – Examples of single-channel target signal upon the occurrence of an event at time $k$. 1) a change of slope, 2) decaying sinusoid with sharp onset 3) two-sided fastly-decaying sinusoid, 4) two-sided fastly-decaying sinusoid on top of a baseline.

which in turn defines a corresponding set $C$.

Given a multi-channel discrete-time signal $y_1, \ldots, y_K \in \mathbb{R}^M$, $M \in \mathbb{N}$, we wish to detect such events and estimate its parameters. For that purpose, we introduce a cost function

$$ J_k(C) \triangleq \overrightarrow{J}_k(C_p) + \overleftarrow{J}_k(C_f) \quad (5.8) $$

to quantify the discrepancy between the observations and a target signal,
assuming the occurrence of an event at time $k$. The cost $\overrightarrow{J}_k(C_p)$ is associated with the “past” model while $\overleftarrow{J}_k(C_f)$ is associated with the “future” one. Since multiple events typically occur within a time interval, the cost $J_k(C)$ will be localized around the time index $k$ of interest. Moreover, since the locations of the events are unknown, the cost $J_k(C)$ needs to be computed and minimized for all $k$. Thus, in Chapter 6, we define several localized cost functions that can be recursively computed. We also address several minimization problems that we often encounter in signal processing applications. Based on these cost functions, we provide likelihood ratios for detecting such events in various conditions (cf. Chapter 7). Finally, the proposed approach is illustrated by some nontrivial applications:

i. single-channel ECG (electrocardiogram) shape detection with integrated baseline removal (cf. Sec. 8.1),

ii. detecting straight-line passes of a magnet and estimating their speed and direction using a 3-axes magnetic field sensor (cf. Sec. 8.2),

iii. detecting and tracing the movement of an electrical dipole from multi-channel voltage measurements (cf. Sec. 8.3),

iv. detection and estimation of a modulated signal (cf. Sec. 8.4),

v. detection and classification of fires (cf. Sec. 8.5),

vi. detection and estimation of model switches (cf. Sec. 8.6).
Chapter 6

Cost Definition, Computation, and Minimization

6.1 Exponentially-Weighted Squared Error Cost

6.1.1 Cost Definition

Given a multi-channel discrete-time signal \( y_1, \ldots, y_K \in \mathbb{R}^M, M \in \mathbb{N} \), and a \( M \)-channel LSSM signal \( f(C) \) with parameters \( \{C, A, s\} \) and unknown \( C \), let \( J_k(C) \) be the exponentially-weighted squared error cost

\[
J_k(C) = \sum_{i=1}^{k} \gamma_{p}^{k-i} \|y_i - f_{i-k}(C)\|^2 + \sum_{i=k+1}^{K} \gamma_{f}^{i-k} \|y_i - f_{i-k}(C)\|^2
\]

(6.1)

\[
= \tilde{J}_k(C_P) + \tilde{J}_k(C_f),
\]

(6.2)

with

\[
\tilde{J}_k(C_P) = \sum_{i=1}^{k} \gamma_{p}^{k-i} \|y_i - C_P A_P^{k-i} s_P\|^2
\]

(6.3)

\[
\tilde{J}_k(C_f) = \sum_{i=k+1}^{K} \gamma_{f}^{i-k} \|y_i - C_f A_f^{i-k} s_f\|^2.
\]

(6.4)
The factors $\gamma_p \in (0, 1]$ and $\gamma_f \in (0, 1]$ effectively limit the horizon of these cost functions within a two-sided exponentially-decaying window around time $k$.

If we wish to replace the standard $L_2$ norm in (6.1) by a weighted $L_2$ norm $\|u\|_{W_Z} = u^T W_Z u$ for some fixed symmetric positive definite matrix $W_Z \in \mathbb{R}^{M \times M}$, we can simply apply the substitution $y_k \leftarrow S_Z y_k$ and $C \leftarrow S_Z C$ with $S_Z$ such that $S_Z^T S_Z = W_Z$ (see also Appendix B).

6.1.2 Recursive Computation of the Cost Function

Both terms in (6.1) can be efficiently computed (for all $k$) as follows. The first term of (6.1) expands into

$$\tilde{J}_k(C_p) = \sum_{i=1}^{k} \gamma_p^{k-i} \|y_i - C_p A_p^{k-i} s_p\|^2 \quad (6.5)$$

$$= \sum_{i=1}^{k} \gamma_p^{k-i} (\|y_i\|^2 - 2 \text{tr} (C_p A_p^{k-i} s_p y_i^T) + \text{tr} (C_p A_p^{k-i} s_p s_p^T (A_p^T)^{k-i} C_p^T)) \quad (6.6)$$

$$= \tilde{\kappa}_k - 2 \text{tr}(C_p \tilde{\xi}_k) + \text{tr}(C_p \tilde{W}_k C_p^T), \quad (6.7)$$

with

$$\tilde{\kappa}_k = \sum_{i=1}^{k} \gamma_p^{k-i} \|y_i\|^2 \in \mathbb{R} \quad (6.8)$$

$$\tilde{\xi}_k = \sum_{i=1}^{k} \gamma_p^{k-i} A_p^{k-i} s_p y_i^T \in \mathbb{R}^{n_p \times M} \quad (6.9)$$

$$\tilde{W}_k = \sum_{i=1}^{k} \gamma_p^{k-i} A_p^{k-i} s_p s_p^T (A_p^T)^{k-i} \in \mathbb{R}^{n_p \times n_p}. \quad (6.10)$$

The quantities (6.8)–(6.10) can be efficiently computed by the forward recursions

$$\tilde{\kappa}_k = \gamma_p \tilde{\kappa}_{k-1} + \|y_k\|^2 \quad (6.11)$$

$$\tilde{\xi}_k = \gamma_p A_p \tilde{\xi}_{k-1} + s_p y_k^T \quad (6.12)$$

$$\tilde{W}_k = \gamma_p A_p \tilde{W}_{k-1}^T A_p^T + s_p s_p^T, \quad (6.13)$$

with the initializations $\tilde{\kappa}_0 = 0$, $\tilde{\xi}_0 = 0$, and $\tilde{W}_0 = 0$. 
Analogously, the second term of (6.1) expands into
\[ \tilde{J}_k(C_f) = \sum_{i=k+1}^{K} \gamma_f^{i-k} \|y_i - C_f A_f^{i-k} s_f\|^2 \] (6.14)
\[ = \tilde{\kappa}_k - 2 \text{tr}(C_f \tilde{\xi}_k) + \text{tr}(C_f \tilde{W}_k C_f^T), \] (6.15)
with
\[ \tilde{\kappa}_k = \sum_{i=k+1}^{K} \gamma_f^{i-k} \|y_i\|^2 \in \mathbb{R} \] (6.16)
\[ \tilde{\xi}_k = \sum_{i=k+1}^{K} \gamma_f^{i-k} A_f^{i-k} s_f y_i^T \in \mathbb{R}^{n_f \times M} \] (6.17)
\[ \tilde{W}_k = \sum_{i=k+1}^{K} \gamma_f^{i-k} A_f^{i-k} s_f^T s_f^T (A_f^T)^{i-k} \in \mathbb{R}^{n_f \times n_f}, \] (6.18)
and the quantities (6.16)–(6.18) can be efficiently computed by the backward recursions
\[ \tilde{\kappa}_k = \gamma_f (\tilde{\kappa}_{k+1} + \|y_{k+1}\|^2) \] (6.19)
\[ \tilde{\xi}_k = \gamma_f A_f (\tilde{\xi}_{k+1} + s_f y_{k+1}^T) \] (6.20)
\[ \tilde{W}_k = \gamma_f A_f (\tilde{W}_{k+1} + s_f s_f^T) A_f^T, \] (6.21)
with the initializations \( \tilde{\kappa}_K = 0, \tilde{\xi}_K = 0, \) and \( \tilde{W}_K = 0. \)

Merging (6.7) and (6.15) in (6.1), we finally have
\[ J_k(C) = \kappa_k - 2 \text{tr}(C \xi_k) + \text{tr}(C W_k C^T), \] (6.22)
with
\[ \kappa_k = \tilde{\kappa}_k + \hat{\kappa}_k \in \mathbb{R} \] (6.23)
\[ \xi_k = \begin{bmatrix} \tilde{\xi}_k \\ \hat{\xi}_k \end{bmatrix} \in \mathbb{R}^{n \times M} \] (6.24)
\[ W_k = \text{diag}(\tilde{W}_k, \hat{W}_k) \in \mathbb{R}^{n \times n}. \] (6.25)

Clearly, \( J_k(C) \) is a general quadratic form in \( C \) and \( J_k(C) \geq 0, \) for all \( C \in \mathbb{R}^{M \times n}. \) Note that \( \tilde{W}_k, \hat{W}_k, \) and \( W_k \) are symmetric positive semi-definite matrices. The quantity \( \xi_k \) depends linearly on the observations \( y_1, \ldots, y_K \) and can be interpreted as the output of \( n \) linear filters.
Besides, the cost in (6.22) can be split channel-wise since

\[ J_k(C) = \sum_{m=1}^{M} \left( \kappa_k^{(m)} - 2C_m \xi_k^{(m)} + C_m W_k C_m^T \right), \] (6.26)

where \( C_m \) is the \( m \)th row of \( C \), \( \xi_k^{(m)} \) the \( m \)th column of \( \xi_k \) (depending on the \( m \)th measurement channel only), and

\[ \kappa_k^{(m)} = \sum_{i=1}^{K} \left( y_i^{(m)} \right)^2, \] (6.27)

with \( y_i^{(m)} \) denoting the \( m \)th element of \( y_i \).

In addition, since \( W_k \) is symmetric positive (semi-)definite, there exists \( S_k \in \mathbb{R}^{n \times n} \) such that \( W_k = S_k^T S_k \). Denoting \( \|P\|_F = \sqrt{\text{tr}(P^T P)} \) the Frobenius norm and assuming \( W_k \) to be invertible, the cost can be written as

\[ J_k(C) = \kappa_k - \text{tr}(\xi_k^T W_k^{-1} \xi_k) + \|S_k(C^T - W_k^{-1} \xi_k)\|_F^2. \] (6.28)

### 6.1.3 Cost Function Minimization

Having computed \( J_k(C) \), we now consider the computation of

\[ \hat{C} = \arg\min_{C \in \mathcal{C}} J_k(C) \] (6.29)

\[ = \arg\min_{\hat{C} \in \tilde{\mathcal{C}}} \kappa_k - 2 \text{tr}(C \xi_k) + \text{tr}(CW_k C^T), \] (6.30)

for some admissible set \( \mathcal{C} \), which amounts to locally (i.e., around time index \( k \)) fitting the model (5.5) to the given data \( y_1, \ldots, y_K \).

In the following, we provide the solution of the minimization (6.30) for many specific sets \( \mathcal{C} \) that appear in signal processing applications.

Table 6.1 summarizes cases of “direct” optimizations with closed-form solutions. Table 6.2 summarizes cases where \( \mathcal{C} \) depends on another set \( \tilde{\mathcal{C}} \) such that the optimization boils down to minimizing a cost \( \tilde{J}_k(\tilde{C}) \) as in (6.22) with modified parameters \( \tilde{\kappa}_k, \tilde{\xi}_k, \) and \( \tilde{W}_k \) and \( \tilde{C} \in \tilde{\mathcal{C}} \). Thus, any specific set \( \mathcal{C} \) of Table 6.2 can be combined through \( \hat{C} \) with any other set from either Table 6.1 or Table 6.2.
6.1 Exponentially-Weighted Squared Error Cost

Unconstrained

Let $\mathcal{C} = \mathbb{R}^{M \times n}$. Then, we have

\[
\hat{C} = (W_k^{-1} \xi_k)^T
\]

\[
\min_{C \in \mathbb{R}^{M \times n}} J_k(C) = \kappa_k - \text{tr}(\xi_k^T W_k^{-1} \xi_k).
\]

Fixed But (Constrained) Scaled

Let $\mathcal{C} = \{\lambda C_1 : \lambda \in \mathbb{R}\}$ where $C_1 \in \mathbb{R}^{M \times n}$ is a fixed matrix. Then, we have

\[
J_k(\lambda C_1) = \kappa_k - 2\lambda \text{tr}(C_1 \xi_k) + \lambda^2 \text{tr}(C_1 W_k C_1^T)
\]

\[
\min_{\lambda \in \mathbb{R}} J_k(\lambda C_1) = \kappa_k - \frac{\text{tr}(C_1 \xi_k)^2}{\text{tr}(C_1 W_k C_1^T)}
\]

\[
\arg\min_{\lambda \in \mathbb{R}} J_k(\lambda C_1) = \frac{\text{tr}(C_1 \xi_k)}{\text{tr}(C_1 W_k C_1^T)}.
\]

Note that this case is easily extended to the cases where $\lambda$ is restricted such that $\lambda \in [a, b]$ or $\lambda \leq a$ or $\lambda \geq a$.

Orthogonal Procrustes Problem (OPP)

Let $\mathcal{C} = \{C \in \mathbb{R}^{M \times n} : C^T C = I_n\}$. $\mathcal{C}$ is non-empty if and only if $M \geq n$. Then, for all $C \in \mathcal{C}$, the cost function simplifies into

\[
J_k(C) = \kappa_k - 2 \text{tr}(C \xi_k) + \text{tr}(W_k).
\]

Therefore,

\[
\hat{C} = \arg\max_{C : C^T C = I_n} \text{tr}(C \xi_k).
\]

This is the well-known Orthogonal Procrustes Problem \[66\].

Using the SVD decomposition

\[
\xi_k = U S V^T,
\]

with $U \in O_n(\mathbb{R})$, $V \in O_M(\mathbb{R})$, and $S \in \mathbb{R}^{n \times M}$ a diagonal matrix with non-negative elements, we have

\[
\hat{C} = \arg\max_{C : C^T C = I_n} \text{tr}(V^T C U S).
\]
As \( \tilde{C} = V^T C U \in \mathbb{R}^{M \times n} \) also satisfies \( \tilde{C}^T \tilde{C} = I_n \), an equivalent optimization problem is

\[
\max_{\tilde{C}: \tilde{C}^T \tilde{C} = I_n} \text{tr} (\tilde{C} S). \tag{6.40}
\]

For such matrix \( \tilde{C} \), we have

\[
\text{tr} (\tilde{C} S) = \sum_{m=1}^{\min(M,n)} s_m \tilde{C}_{m,m} \leq \sum_{m=1}^{n} s_m = \text{tr} (S), \tag{6.41}
\]

\( s_m \) being the singular values and \( |\tilde{C}_{m,m}| \leq 1 \) since the columns of \( \tilde{C} \) are of unit norm. Moreover equality holds with \( \tilde{C} = I_{M,n} \) (pseudo identity matrix of size \( M \times n \)). Thus, we have

\[
\hat{C} = V I_{M,n} U^T \tag{6.42}
\]

with the SVD decomposition \( \xi_k = U S V^T \), and \( \|P\|_* \) denotes the nuclear norm of \( P \), that is the sum of its singular values.

**Wahba’s Problem**

Let \( M = n \) and \( \mathcal{C} = \{ C \in \mathbb{R}^{n \times n} : C^T C = I_n, |C| = 1 \} = SO_n(\mathbb{R}) \). This set excludes the orthogonal matrices of determinant \(-1\) (i.e., we are looking for a rotation matrix).

Similarly to the “Orthogonal Procrustes Problem” we have

\[
\hat{C} = \arg\max_{C \in SO_n(\mathbb{R})} \text{tr}(C \xi_k). \tag{6.44}
\]

This is known as Wahba’s problem \([76]\). The solution is given by the Kabsch algorithm \([34,52]\). From the SVD decomposition \( \xi_k = U S V^T \) we have

\[
\hat{C} = V \tilde{S} U^T, \tag{6.45}
\]

with \( \tilde{S} \) of the same diagonal form as \( S \) where all singular values are replaced by one except for the minimum singular value \( s_{\text{min}} \) that is replaced by \( |V U^T| \). As a consequence, we indeed have \( |\hat{C}| = 1 \) and

\[
\max_{C \in SO_n(\mathbb{R})} \text{tr}(C \xi_k) = \|\xi_k\|_* + (|V U^T| - 1)s_{\text{min}}. \tag{6.46}
\]
6.1 Exponentially-Weighted Squared Error Cost

Scaled Orthogonal Procrustes Problem

Let \( C = \{ \lambda O : \lambda \in \mathbb{R}_+, \ O \in \mathbb{R}^{M \times n}, \ O^T O = I_n \} \). Then, we have

\[
J_k(\lambda O) = \kappa_k - 2\lambda \text{tr}(O \xi_k) + \lambda^2 \text{tr}(W_k). \tag{6.47}
\]

As \( \lambda \geq 0 \), the optimization over \( O \) does not depend on \( \lambda \) and is done as in the “Orthogonal Procrustes Problem” (cf. Sec. 6.1.3). Then, the optimization over \( \lambda \in \mathbb{R}_+ \) is similar to the case of “Fixed But Scaled” (cf. Sec. 6.1.3). Therefore, using the SVD decomposition \( \xi_k = U S V^T \), we have

\[
\arg\min_{O: \ O^T O=I_n} J_k(\lambda O) = V I_{M,n} U^T \tag{6.48}
\]

\[
\hat{\lambda} = \frac{\|\xi_k\|_*}{\text{tr}(W_k)} \tag{6.49}
\]

\[
\min_{\lambda \in \mathbb{R}_+, \ O^T O=I_n} J_k(\lambda O) = \kappa_k - \frac{\|\xi_k\|_*^2}{\text{tr}(W_k)}. \tag{6.50}
\]

Note that indeed \( \hat{\lambda} \geq 0 \) since \( \|\xi_k\|_* \) is a sum of non-negative (singular) values and \( W_k \) is symmetric positive definite.

Scaled Wahba’s Problem

Let \( M = n \) and \( C = \{ \lambda O : \lambda \in \mathbb{R}_+, \ O \in SO_n(\mathbb{R}) \} \). It is straightforward to adapt the “Scaled Orthogonal Procrustes Problem” with “Wahba’s Problem”. The rotation matrix \( O \) is found as in “Wahba’s Problem” while the positive scalar \( \lambda \) and the minimum cost is found as in (6.49) and (6.50) by replacing \( \|\xi_k\|_* \) by \( \|\xi_k\|_* + (|VU^T| - 1)s_{\min} \).

Weighted Orthogonal Procrustes Problem (WOPP)

Let \( P \in \mathbb{R}^{M \times n} \) be a fixed matrix. Let \( C = \{ PO : O \in O_n(\mathbb{R}) \} \).

According to (6.28), the minimization problem is equivalent to finding

\[
\hat{O} = \arg\min_{O \in O_n(\mathbb{R})} \|S_k(O^T P^T - W_k^{-1} \xi_k)\|_F^2 \tag{6.51}
\]

\[
= \arg\min_{O \in O_n(\mathbb{R})} \text{tr} \left( (PO - \tilde{\xi}_k) W_k (PO - \tilde{\xi}_k)^T \right), \tag{6.52}
\]

with \( \tilde{\xi}_k = \xi_k^T W_k^{-1} \). As \( W_k \) is a symmetric positive definite matrix, it can be diagonalized in an orthonormal basis such that \( W_k = U \Lambda U^T \) with
\( U \in \mathcal{O}_n(\mathbb{R}) \) and \( \Lambda \in \mathbb{R}^{n \times n} \) is a diagonal matrix with positive entries. Then, we have

\[
\hat{O}U = \arg\min_{V \in \mathcal{O}_n(\mathbb{R})} \text{tr} \left( (PV - \xi')\Lambda(PV - \xi')^T \right),
\]

(6.53)

with \( V = OU \in \mathcal{O}_n(\mathbb{R}) \) and \( \xi' = \xi_k^TW_k^{-1}U \in \mathbb{R}^{M \times n} \).

This optimization problem is called the weighted orthogonal procrustes problem (WOPP) \([40,41]\). In the general case (i.e., when it does not reduce to an “Orthogonal Procrustes Problem”), there is no closed-form solution. However, an iterative algorithm proposed in \([41]\) finds a reasonable estimate. It works as follows.

1. Find \( \rho \in \mathbb{R} \) such that \( \rho^2I - P^TP \in \mathbb{R}^{n \times n} \) is symmetric positive definite and define \( P_a \in \mathbb{R}^{n \times n} \) such that \( P_aP_a^T = \rho^2I - PP^T \) (e.g., using a Cholesky decomposition). The matrix \( P^* = \begin{bmatrix} P \\ P_a \end{bmatrix} \in \mathbb{R}^{(M+n) \times n} \) is such that \( (P^*)^TP^* = \rho^2I \).

2. Initialize \( V \in \mathcal{O}_n(\mathbb{R}) \) (e.g., \( V = I \) or by solving the unweighted problem pretending that \( \Lambda = I \)) and set \( Y^* = \begin{bmatrix} \xi' \\ P_aV \end{bmatrix} \in \mathbb{R}^{(M+n) \times n} \).

3. Repeat the following steps: Compute the SVD decomposition \( \Lambda(Y^*)^TX^* = QSF^T \) and set \( V = FQ^T \) and \( Y^* = \begin{bmatrix} \xi' \\ P_aV \end{bmatrix} \).

A special case of interest is when \( C = \{PO : O \in SO_3(\mathbb{R})\} \). An alternative algorithm is proposed in \([73]\).

**Observation Matrix on the Unit Sphere**

Let \( C = \{C \in \mathbb{R}^{M \times n} : \|C\|_F = 1\} \), where \( \|C\|_F = \sqrt{\text{tr}(C^TC)} \) denotes the Frobenius norm. The Lagrangian of this optimization problem is

\[
L(C, \lambda) = \kappa_k - 2\text{tr}(C\xi_k) + \text{tr}(CW_kC^T) - \lambda \left( \text{tr}(C^TC) - 1 \right).
\]

(6.54)

Setting the gradient of the Lagrangian to zero leads to

\[
\hat{C} = \xi_k^T(W_k - \lambda I)^{-1},
\]

(6.55)

with \( \lambda \) satisfying

\[
\text{tr} \left( (W_k - \lambda I)^{-2} \xi_k\xi_k^T \right) = 1.
\]

(6.56)
Since $W_k$ is a symmetric positive definite matrix, it can be diagonalized in an orthonormal basis such that $W_k = U \text{diag}(\lambda_1, \ldots, \lambda_n)U^T$ with $U \in \mathcal{O}_n(\mathbb{R})$ and $\lambda_i > 0$, $i \in \{1, \ldots, n\}$, being in increasing order. Then, (6.56) becomes
\[
\sum_{i=1}^{n} \frac{(U^T \xi_k \xi_k^T U)_{i,i}^2}{(\lambda_i - \lambda)^2} = 1. \tag{6.57}
\]
This last equation is the so-called secular equation. In [30], an iterative algorithm is proposed to estimate $\lambda$ satisfying such an equation:

1. Initialize $\lambda^{(0)} = \lambda_{i_0} - |\{U^T \xi_k \xi_k^T U\}_{i_0,i_0}|$ with $i_0$ being the smallest index such that $\{U^T \xi_k \xi_k^T U\}_{i_0,i_0} \neq 0$.
2. Repeat
   \[
   \lambda^{(i+1)} = \lambda^{(i)} - 2 \frac{f(\lambda^{(i)})}{f'(\lambda^{(i)})} \left( \sqrt{f(\lambda^{(i)})} - 1 \right), \tag{6.58}
   \]
   with
   \[
   f(\lambda) = \sum_{i=1}^{n} \frac{(U^T \xi_k \xi_k^T U)_{i,i}^2}{(\lambda_i - \lambda)^2}, \tag{6.59}
   \]
   while $\lambda^{(i+1)} < \lambda^{(i)}$.

In theory, this algorithm produces a strictly decreasing sequence of approximations $\lambda^{(i)}$ [30].

**Post-Multiplied**

Let $d \in \mathbb{N}$ and $P \in \mathbb{R}^{d \times n}$ be some fixed matrix. Let $\mathcal{C} = \{\tilde{C}P : \tilde{C} \in \tilde{C} \subset \mathbb{R}^{M \times d}\}$. Then, we have
\[
\tilde{J}_k(\tilde{C}) = J_k(\tilde{C}P) \tag{6.60}
= \kappa_k - 2 \text{tr}(\tilde{C}(P\xi_k)) + \text{tr}(\tilde{C}(PW_kP^T)\tilde{C}^T). \tag{6.61}
\]
Thus, minimizing $\tilde{J}_k(\tilde{C})$ is again a problem of the form (6.30) with
\[
\tilde{\kappa}_k = \kappa_k \tag{6.62}
\]
\[
\tilde{\xi}_k = P\xi_k \tag{6.63}
\]
\[
\tilde{W}_k = PW_kP^T. \tag{6.64}
\]
For instance, if $\mathcal{C} = \{\tilde{C}P : \tilde{C} \in \mathbb{R}^{M \times d}\}$, then the rows of $C$ lies in the vector space spanned by the rows of $P$. 
Fixed Offset

Let $\mathcal{C} = \{C_0 + \tilde{C} : \tilde{C} \in \tilde{\mathcal{C}} \subset \mathbb{R}^{M \times n}\}$ where $C_0 \in \mathbb{R}^{M \times n}$ is a fixed matrix. Then, we have

$$\tilde{J}_k(\tilde{C}) = J_k(C_0 + \tilde{C}) = \kappa_k - 2 \text{tr}(C_0 \xi_k) + \text{tr}(C_0 W_k C_0^T)$$
$$-2 \text{tr}(\tilde{C}(\xi_k - W_k C_0^T)) + \text{tr}(\tilde{C} W_k \tilde{C}^T).$$

Thus, minimizing $\tilde{J}_k(\tilde{C})$ is again a problem of the form (6.30) with

$$\tilde{\kappa}_k = \kappa_k - 2 \text{tr}(C_0 \xi_k) + \text{tr}(C_0 W_k C_0^T)$$
$$\tilde{\xi}_k = \xi_k - W_k C_0^T$$
$$\tilde{W}_k = W_k.$$

Constrained Offset

Let $d \in \mathbb{N}$ and $P \in \mathbb{R}^{d \times n}$ be a fixed matrix. Let $\mathcal{C} = \{\tilde{C} + Q \tilde{P} : Q \in \mathbb{R}^{M \times d}, \tilde{C} \in \tilde{\mathcal{C}}\}$. Then, we have

$$J_k(\tilde{C} + Q \tilde{P}) = \kappa_k - 2 \text{tr}(Q \tilde{P}(\xi_k - W_k \tilde{C}^T))$$
$$-2 \text{tr}(\tilde{C} \tilde{P}(\tilde{C} - W_k \tilde{P}^T \tilde{C}^T))$$
$$+ \text{tr}(Q \tilde{P} W_k \tilde{P}^T \tilde{C}^T) + \text{tr}(\tilde{C} W_k \tilde{C}^T).$$

The minimization over $Q \in \mathbb{R}^{M \times d}$ leads to

$$\hat{Q} = (\xi_k^T - \tilde{C} W_k) \tilde{P}^T (P W_k \tilde{P}^T)^{-1},$$

and

$$\hat{J}_k(\tilde{C}) = \min_{Q \in \mathbb{R}^{M \times d}} J_k(\tilde{C} + Q \tilde{P})$$
$$= \kappa_k - \text{tr}(\xi_k^T \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P \xi_k)$$
$$-2 \text{tr}(\tilde{C}(\xi_k - W_k \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P \xi_k))$$
$$+ \text{tr}(\tilde{C}(W_k - W_k \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P W_k) \tilde{C}^T).$$

Thus, minimizing $\hat{J}_k(\tilde{C})$ is again a problem of the form (6.30) with

$$\tilde{\kappa}_k = \kappa_k - \text{tr}(\xi_k^T P \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P \xi_k)$$
$$\tilde{\xi}_k = \xi_k - W_k \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P \xi_k$$
$$\tilde{W}_k = W_k - W_k \tilde{P}^T (P W_k \tilde{P}^T)^{-1} P W_k.$$
6.1 Exponentially-Weighted Squared Error Cost

**Linear Combination**

Let \( \{ C_i \in \mathbb{R}^{M \times n} : i \in \{1, \ldots, d\} \} \) be a given set of \( d \in \mathbb{N} \) matrices. Let \( \mathcal{C} = \{ \sum_{i=1}^{d} \tilde{c}_i C_i : \tilde{C} = [\tilde{c}_1, \ldots, \tilde{c}_d] \in \tilde{C} \subset \mathbb{R}^{1 \times d} \} \). For \( d = 1 \) and \( \tilde{C} = \mathbb{R} \), this is the “Fixed But Scaled” problem.

Then, we have

\[
\tilde{J}_k(\tilde{C}) = J_k \left( \sum_{i=1}^{d} \tilde{c}_i C_i \right) 
\]

(6.77)

\[
= \kappa_k - 2 \sum_{i=1}^{d} \tilde{c}_i \text{tr}(C_i \xi_k) + \sum_{i=1}^{d} \sum_{j=1}^{d} \tilde{c}_i \tilde{c}_j \text{tr} \left( C_i W_k C_j^T \right) 
\]

(6.78)

with

\[
\{ \tilde{\xi}_k \}_i = \text{tr}(C_i \xi_k), \ \forall i \in \{1, \ldots, d\} \]

(6.79)

\[
\{ \tilde{W}_k \}_{i,j} = \text{tr}(C_i W_k C_j^T), \ \forall (i,j) \in \{1, \ldots, d\}^2. 
\]

(6.80)

Thus, minimizing \( \tilde{J}_k(\tilde{C}) \) is again a problem of the form (6.30). Note that \( \tilde{W}_k \) is symmetric positive semi-definite since for all \( \tilde{C} \in \mathbb{R}^d \), \( \tilde{C} \tilde{W}_k \tilde{C}^T = \text{tr}(C W_k C^T) \geq 0 \) (with \( C = \sum_{i=1}^{d} \tilde{c}_i C_i \)).

**Pre-Multiplied**

Let \( d \in \mathbb{N} \) and \( P \in \mathbb{R}^{M \times d} \) be a fixed matrix. Let \( \mathcal{C} = \{ P \tilde{C} : \tilde{C} \in \tilde{C} \subset \mathbb{R}^{d \times n} \} \). Then, we have

\[
\tilde{J}_k(\tilde{C}) = J_k (P \tilde{C}) 
\]

(6.81)

\[
= \kappa_k - 2 \text{tr}(\tilde{C} \xi_k P) + \text{tr}(\tilde{C} W_k \tilde{C}^T P^T P) 
\]

(6.82)

\[
= \kappa_k - 2 \text{vec}(\tilde{C}^T)^T \text{vec}(\xi_k P) + \text{vec}(\tilde{C}^T)^T (P^T P \otimes W_k) \text{vec}(\tilde{C}^T), 
\]

(6.83)

where we have used (A.3).

Thus, minimizing \( J_k(\tilde{C}) \) is a problem of the form (6.30) with respect to \( \text{vec}(\tilde{C}^T)^T \in \text{vec}(\tilde{C}^T)^T \subset \mathbb{R}^{1 \times (dn)} \) with

\[
\tilde{\kappa}_k = \kappa_k 
\]

(6.84)

\[
\tilde{\xi}_k = \text{vec}(\xi_k P) 
\]

(6.85)

\[
\tilde{W}_k = P^T P \otimes W_k. 
\]

(6.86)
but not with respect to $\tilde{C}$ directly. In particular, the cost $\tilde{J}_k(\tilde{C})$ does not necessarily split in a sum of costs, where each cost exclusively depends on one row of $\tilde{C}$. This splitting only happens when $P^TP$ is a diagonal matrix.

Note that replacing the L2 norm by a weighted L2 norm in (6.1) inherently leads to a case of “Pre-Multiplied” observation matrix.
### Table 6.1 – Direct Optimizations with Closed Form

<table>
<thead>
<tr>
<th></th>
<th>Unconstrained</th>
<th>Fixed But Scaled</th>
<th>Orthogonal Procrustes Problem</th>
<th>Scaled Orthogonal Procrustes Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>$\mathbb{R}^{M \times n}$</td>
<td>${\lambda C : \lambda \in \mathbb{R}}$</td>
<td>${C \in \mathbb{R}^{M \times n} : C^T C = I_n}$</td>
<td>${\lambda O : \lambda \in \mathbb{R}_+, O \in \mathbb{R}^{M \times n}, O^T O = I_n}$</td>
</tr>
<tr>
<td>$\argmin_{C \in \mathcal{C}} J_k(C)$</td>
<td>$\frac{\operatorname{tr}(C_1 \xi_k) W_k^{-1} C_1}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\frac{\operatorname{tr}(C_1 \xi_k) W_k^{-1} C_1}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\frac{\operatorname{tr}(C_1 \xi_k) W_k^{-1} C_1}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\frac{\operatorname{tr}(C_1 \xi_k) W_k^{-1} C_1}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
</tr>
<tr>
<td>$\min_{C \in \mathcal{C}} J_k(C)$</td>
<td>$\kappa_k - \frac{\operatorname{tr}(C_1 \xi_k)^2}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\kappa_k - \frac{\operatorname{tr}(C_1 \xi_k)^2}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\kappa_k - \frac{\operatorname{tr}(C_1 \xi_k)^2}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
<td>$\kappa_k - \frac{\operatorname{tr}(C_1 \xi_k)^2}{\operatorname{tr}(C_1 W_k C_1^T)}$</td>
</tr>
</tbody>
</table>

with SVD decomposition $\xi_k = USV^T$ for the optimum $Q$ see (6.71).

### Table 6.2 – Nested Optimizations

<table>
<thead>
<tr>
<th></th>
<th>Post-Multiplied</th>
<th>Fixed Offset</th>
<th>Constrained Offset</th>
<th>Linear Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given $P \in \mathbb{R}^{M \times d}$</td>
<td>$C_0 \in \mathbb{R}^{M \times n}$</td>
<td>$P \in \mathbb{R}^{d \times n}$</td>
<td>${C_i \in \mathbb{R}^{M \times n} : i \in {1, \ldots, d}}$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>$\mathcal{C}P : \mathcal{C} \in \mathcal{C}$</td>
<td>$\mathcal{C} + C_0 : \mathcal{C} \in \mathcal{C}$</td>
<td>$\mathcal{C} + QP : \mathcal{C} \in \mathcal{C}, Q \in \mathbb{R}^{M \times d}$</td>
<td>${\sum_{i=1}^{d} \tilde{C}_i C_i : \tilde{C} = [\tilde{C}_1, \ldots, \tilde{C}_d] \in \mathcal{C} \subset \mathbb{R}^{1 \times d}}$</td>
</tr>
<tr>
<td>$J_k(\tilde{C})$</td>
<td>$J_k(\mathcal{C}P)$</td>
<td>$J_k(\mathcal{C} + C_0)$</td>
<td>$\min_{Q \in \mathbb{R}^{M \times d}} J_k(\mathcal{C} + QP)$</td>
<td>$J_k(\sum_{i=1}^{d} \tilde{C}_i C_i)$</td>
</tr>
<tr>
<td>$\tilde{\kappa}_k$</td>
<td>$\kappa_k$</td>
<td>$\kappa_k - 2 \operatorname{tr}(C_0 \xi_k) + \operatorname{tr}(C_0 W_k C_0^T)$</td>
<td>$\kappa_k - \operatorname{tr}(\xi_k^T P^T (PW_k P^T)^{-1} P \xi_k)$</td>
<td>$\kappa_k$</td>
</tr>
<tr>
<td>$\tilde{\xi}_k$</td>
<td>$P \xi_k$</td>
<td>$\xi_k - W_k C_0^T$</td>
<td>$\xi_k - W_k P^T (PW_k P^T)^{-1} P \xi_k$</td>
<td>${\operatorname{tr}(C_i \xi_k)}_{i \in {1, \ldots, d}} \in \mathbb{R}^d$</td>
</tr>
<tr>
<td>$\tilde{W}_k$</td>
<td>$PW_k P^T$</td>
<td>$W_k$</td>
<td>$W_k - W_k P^T (PW_k P^T)^{-1} PW_k$</td>
<td>${\operatorname{tr}(C_i W_k C_j^T)}_{i,j \in {1, \ldots, d}} \in \mathbb{R}^{d \times d}$</td>
</tr>
</tbody>
</table>

(*) for the optimum $Q$ see [6.71].
6.2 LSSM-Weighted Squared Error Cost

Thanks to the exponential window introduced via $\gamma_p$ in (6.3) and $\gamma_f$ in (6.4), the estimation of $C \in \mathcal{C}$ by minimizing $J_k(C)$ is also localized. Effectively, only the observed samples around time index $k$ influence the cost function, and thus the estimation. In particular, this localization reveals to be useful and needed when several successive signal shapes from $\mathcal{C}$ occur within the observed signal and for online algorithms since the number of samples indefinitely grows ($K \to \infty$). In fact, the exponential window can be replaced by a LSSM window while still preserving a recursive cost computation.

6.2.1 Cost Definition

Given a multi-channel discrete-time signal $y_1, \ldots, y_K \in \mathbb{R}^M$, $M \in \mathbb{N}$, and a $M$-channel LSSM signal $f(C)$ with parameters $\{C, A, s\}$ and unknown $C$, let $J_k(C)$ be the LSSM-weighted squared error cost

$$J_k(C) = \sum_{i=1}^{K} w_{i-k} \| y_i - f_{i-k}(C) \|^2$$

(6.87)

$$= \tilde{J}_k(C_p) + \tilde{J}_k(C_f),$$

(6.88)

with

$$\tilde{J}_k(C_p) = \sum_{i=1}^{k} w_{i-k} \| y_i - C_p A_p^{k-i} s_p \|^2$$

(6.89)

$$\tilde{J}_k(C_f) = \sum_{i=k+1}^{K} w_{i-k} \| y_i - C_f A_f^{i-k} s_f \|^2,$$

(6.90)

where $w_{i-k}$ is a LSSM signal with given parameters $\{C_w, A_w, s_w\}$ and shifted by a time $k$, i.e.,

$$w_{i-k} = \begin{cases} C_{p,w} A_{p,w}^{k-i} s_{p,w}, & \text{for } i \leq k \\ C_{f,w} A_{f,w}^{i-k} s_{f,w}, & \text{for } i > k. \end{cases}$$

(6.91)

For a proper localization of the cost, we further require that $w_j \in \mathbb{R}$, $j \in \mathbb{Z}$, is a LSSM signal such that $|w_j| \to 0$ as $j \to \pm \infty$ and $\sum_{j \in \mathbb{Z}} |w_j| < +\infty$. If needed, we can also ensure non-negative window weights by squaring the LSSM window in (6.91) which is again a LSSM window by Proposition 4.
The set of LSSM windows is quite large as demonstrated by Proposition 6. In particular, any finite duration window, and thus a rectangular window, can be obtained by choosing \( A_{p,w} \) and \( A_{f,w} \) to be nilpotent. Furthermore, since the two-sided exponential window introduced in (6.3) and (6.4) with \( \gamma_p \) and \( \gamma_f \) are generated by a LSSM of order one and with \( C_{p,w} = s_{p,w} = C_{f,w} = s_{f,w} = 1 \), \( A_{p,w} = \gamma_p \), and \( A_{f,w} = \gamma_f \), the exponentially-weighted squared error cost in (6.1) is a special case of the LSSM-weighted squared error cost (6.87).

Note that it is possible to replace the standard \( L^2 \) norm in (6.87) by a weighted \( L^2 \) norm \( \|u\|_{W,Z} = u^T W Z u \) for some matrix \( W, Z \in S^*_+ \). In addition, by allowing \( W, Z \in S^*_+ \) to be unknown, it actually regularizes the estimation of \( C \), as explained in Appendix B.

6.2.2 Recursive Computation of the Cost Function

Essentially, recursive cost computations are preserved since the product of LSSM signals is again a LSSM signal (cf. Proposition 4). We now make those recursions explicit.

Both terms in (6.87) can be recursively and efficiently computed. Expanding (6.89) as in (6.6), we get

\[
\tilde{J}_k(C_p) = C_{p,w} \sum_{i=1}^k A_{p,w}^{k-i} s_{p,w} \|y_i\|^2 \\
-2 \text{tr} \left( \sum_{i=1}^k C_{p,w} A_{p,w}^{k-i} s_{p,w} C_p A_p^{k-i} s_p y_i^T \right) \\
+ \sum_{i=1}^k C_{p,w} A_{p,w}^{k-i} s_{p,w} C_p A_p^{k-i} s_p s_p^T (A_p^T)_{k-i} C_p^T \right) .
\]

Then, using Proposition 4 easily extended to the case where “\( C_1 \)” is a matrix, we get, for all \( j \in \mathbb{N}_0 \),

\[
C_{p,w} A_{p,w}^j s_{p,w} C_p A_p^j s_p = (C_p \otimes C_{p,w})(A_p \otimes A_{p,w})^j (s_p \otimes s_{p,w}),
\]

from which we obtain

\[
\tilde{J}_k(C_p) = C_{p,w} \tilde{\chi}_k - 2 \text{tr} \left( (C_p \otimes C_{p,w}) \tilde{\zeta}_k \right) + \text{tr} \left( (C_p \otimes C_{p,w}) \tilde{S}_k C_p^T \right),
\]

(6.94)
with

\[ \tilde{\chi}_k = \sum_{i=1}^{k} A_{p,w}^{k-i} s_{p,w} \| y_i \|^2 \]  
(6.95)

\[ \tilde{\zeta}_k = \sum_{i=1}^{k} (A_p \otimes A_{p,w})^{k-i} (s_p \otimes s_{p,w}) y_i^T \]  
(6.96)

\[ \tilde{S}_k = \sum_{i=1}^{k} (A_p \otimes A_{p,w})^{k-i} (s_p \otimes s_{p,w}) s_p^T (A_p^T)^{k-i}. \]  
(6.97)

The quantities (6.95)–(6.97) can be efficiently computed by the forward recursions

\[ \tilde{\chi}_k = A_{p,w} \tilde{\chi}_{k-1} + s_{p,w} \| y_k \|^2 \]  
(6.98)

\[ \tilde{\zeta}_k = (A_p \otimes A_{p,w}) \tilde{\zeta}_{k-1} + (s_p \otimes s_{p,w}) y_k^T \]  
(6.99)

\[ \tilde{S}_k = (A_p \otimes A_{p,w}) \tilde{S}_{k-1} A_p^T + (s_p \otimes s_{p,w}) s_p^T, \]  
(6.100)

initialized with zeros at \( k = 0 \).

Analogously, the second term in (6.87) expands into

\[ \tilde{J}_k(C_f) = C_{f,w} \tilde{\chi}_k - 2 \text{tr} \left( (C_f \otimes C_{f,w}) \tilde{\zeta}_k \right) + \text{tr} \left( (C_f \otimes C_{f,w}) \tilde{S}_k C_f^T \right), \]  
(6.101)

with

\[ \tilde{\chi}_k = \sum_{i=k+1}^{K} A_{f,w}^{i-k} s_{f,w} \| y_i \|^2 \]  
(6.102)

\[ \tilde{\zeta}_k = \sum_{i=k+1}^{K} (A_f \otimes A_{f,w})^{i-k} (s_f \otimes s_{f,w}) y_i^T \]  
(6.103)

\[ \tilde{S}_k = \sum_{i=k+1}^{K} (A_f \otimes A_{f,w})^{i-k} (s_f \otimes s_{f,w}) s_f^T (A_f^T)^{i-k}, \]  
(6.104)

and the quantities (6.102)–(6.104) can be efficiently computed by the backward recursions

\[ \tilde{\chi}_k = A_{f,w} \left( \tilde{\chi}_{k+1} + s_{f,w} \| y_{k+1} \|^2 \right) \]  
(6.105)

\[ \tilde{\zeta}_k = (A_f \otimes A_{f,w}) \left( \tilde{\zeta}_{k+1} + (s_f \otimes s_{f,w}) y_{k+1}^T \right) \]  
(6.106)

\[ \tilde{S}_k = (A_f \otimes A_{f,w}) \left( \tilde{S}_{k+1} + (s_f \otimes s_{f,w}) s_f^T \right) A_f^T, \]  
(6.107)
initialized with zeros at $k = K$.

Actually, the cost in (6.94) can be written as in (6.7), i.e.,

$$
\overrightarrow{J}_k(C_p) = \overrightarrow{\kappa}_k - 2 \text{tr}(C_p \overrightarrow{\xi}_k) + \text{tr}(C_p \overrightarrow{W}_k C_p^T),
$$

(6.108)

with, for $(i, j) \in \{1, \ldots, n_p\}^2$ and $j' \in \{1, \ldots, M\}$,

$$
\overrightarrow{\kappa}_k = C_{p, w} \overrightarrow{\chi}_k,
$$

(6.109)

$$
\{ \overrightarrow{\xi}_k \}_{i, j'} = \text{tr} \left( (P_{j', i}^{(M, n_p)} \otimes C_{p, w}) \overrightarrow{\xi}_k \right),
$$

(6.110)

$$
\{ \overrightarrow{W}_k \}_{i, j} = \text{tr} \left( (P_{j, i}^{(n_p, n_p)} \otimes C_{p, w}) \overrightarrow{S}_k \right),
$$

(6.111)

where $P_{i, j}^{(m, r)}$ is the $m \times r$ matrix being 1 at index $(i, j)$ and zero anywhere else. The proofs follow from the decomposition

$$
C_p = \sum_{i, j} \{ C_p \}_{i, j} P_{i, j}^{(M, n_p)}.
$$

(6.112)

Furthermore, plugging (6.96) and (6.97) respectively in (6.110) and (6.111), we have the relations

$$
\overrightarrow{\xi}_k = \sum_{i=1}^{k} (C_{p, w} A_{p, w}^{k-i} s_{p, w}) A_{p}^{k-i} s_{p} y_i^T \in \mathbb{R}^{n_p \times M}
$$

(6.113)

$$
\overrightarrow{W}_k = \sum_{i=1}^{k} (C_{p, w} A_{p, w}^{k-i} s_{p, w}) A_{p}^{k-i} s_{p} (A_{p}^T)^{k-i} \in \mathbb{R}^{n_p \times n_p},
$$

(6.114)

which implies that $\overrightarrow{W}_k$ is symmetric but not necessarily positive semi-definite (unless the LSSM window weights are non-negative).

Analogously, the cost in (6.101) writes as in (6.15), i.e.,

$$
\overrightarrow{J}_k(C_f) = \overrightarrow{\kappa}_k - 2 \text{tr}(C_f \overrightarrow{\xi}_k) + \text{tr}(C_f \overrightarrow{W}_k C_f^T),
$$

(6.115)

with, for $(i, j) \in \{1, \ldots, n_f\}^2$ and $j' \in \{1, \ldots, M\}$,

$$
\overrightarrow{\kappa}_k = C_{f, w} \overrightarrow{\chi}_k,
$$

(6.116)

$$
\{ \overrightarrow{\xi}_k \}_{i, j'} = \text{tr} \left( (P_{j', i}^{(M, n_f)} \otimes C_{f, w}) \overrightarrow{\xi}_k \right),
$$

(6.117)

$$
\{ \overrightarrow{W}_k \}_{i, j} = \text{tr} \left( (P_{j, i}^{(n_f, n_f)} \otimes C_{f, w}) \overrightarrow{S}_k \right).
$$

(6.118)
Furthermore, plugging (6.103) and (6.104) respectively in (6.117) and (6.118), we have the relations

$$\xi_k = \sum_{i=k+1}^{K} (C_f, w A_f^{i-k} s_f, w) A_f^{i-k} s_f y_i^T \in \mathbb{R}^{n_{f} \times M}$$

(6.119)

$$\hat{W}_k = \sum_{i=k+1}^{K} (C_f, w A_f^{i-k} s_f, w) A_f^{i-k} s_f (A_f^T)^{i-k} \in \mathbb{R}^{n_{f} \times n_{f}}$$

(6.120)

which implies that $\hat{W}_k$ is symmetric but not necessarily positive semidefinite (unless the LSSM window weights are non-negative).

Thus, the LSSM-weighted squared error cost (6.87) has the same form as (6.22), i.e., writes as

$$J_k(C) = \kappa_k - 2 \text{tr}(C\xi_k) + \text{tr}(CW_kC^T),$$

(6.121)

with $\kappa_k = \kappa_k + \kappa_k$, $\xi_k^T = \begin{bmatrix} \xi_k^T & \xi_k^T \end{bmatrix}$, and $W_k = \text{diag}(\hat{W}_k, \hat{W}_k)$, but the symmetric matrix $W_k$ is not guaranteed to be positive semi-definite anymore.

Since the two-sided exponential window introduced in (6.1) is a LSSM window, the recursions derived in Sec. 6.1.2 are a special case of the recursions derived in this section, where the quantities $\chi_k$, $\zeta_k$, $S_k$, $\bar{\chi}_k$, $\hat{\zeta}_k$, $\hat{S}_k$ coincide with $\kappa_k$, $\xi_k$, $\hat{W}_k$, $\kappa_k$, $\hat{\xi}_k$, $\hat{W}_k$, respectively.

### 6.2.3 Cost Function Minimization

Since the LSSM-weighted squared error cost (6.87) has the same form as (6.22) where $W_k$ is symmetric but not necessarily positive semi-definite, Sec. 6.1.3 also applies for minimizing (6.87), up to the adaptation of $W_k$ not necessarily positive semi-definite.

### 6.2.4 Locality, Stability, and Steady-State

The forward recursions (6.98) and (6.99) involving the measurements $y_1, \ldots, y_K$ can be expressed as

$$X_{k+1} = AX_k + U_k,$$

(6.122)

$k \geq 0$, for some input sequence $U_k$ depending on the measurements, which can be seen as a LSSM with time-dependent inputs. Note that
the symbols $X_k$ and $A$ are here freshly defined. For numerical stability, the unforced system

$$X_{k+1} = AX_k$$  \hfill (6.123)

must be stable, that is to say, for any initial state $X_0$, $X_k$ must go to zero as $k \to +\infty$. This is guaranteed if $A$ is stable, i.e., all its eigenvalues are inside the open unit disc in the complex plane. As a result, any numerical error made at a given time index will exponentially decrease over time, which ensures numerical stability of the recursions (i.e., exponential stability of the dynamical system).

On the other hand, the forward recursion (6.100) independent of the measurements is of the form

$$V_{k+1} = AV_k \tilde{A}^T + Q,$$  \hfill (6.124)

for $k \geq 0$. Vectorizing (6.124) using (A.2) leads to

$$\text{vec}(V_{k+1}) = (\tilde{A} \otimes A) \text{vec}(V_k) + \text{vec}(Q),$$  \hfill (6.125)

which is also a LSSM but with a constant input. It follows that when $\tilde{A} \otimes A$ is stable, $V_k \to V_{ss}$ as $k \to +\infty$ and

$$V_{ss} = \sum_{k=0}^{+\infty} A^k Q (\tilde{A}^T)^k.$$  \hfill (6.126)

$V_{ss}$ is called a steady-state matrix and is solution of

$$V_{ss} = AV_{ss} \tilde{A}^T + Q,$$  \hfill (6.127)

or equivalently

$$(I - \tilde{A} \otimes A) \text{vec}(V_{ss}) = \text{vec}(Q).$$  \hfill (6.128)

Note that when (6.127) can be written such that $A = \tilde{A}$, this is known as a Lyapunov equation \cite[Appendix D]{35}.

Consequently, instead of updating $V_k$ as in (6.124), the steady-state matrix can be used as a substitute. This reduces the computations required at each iteration and creates a mismatch only for the first sample indices when the assumption $V_k \approx V_{ss}$ is not valid.

In a nutshell, if $A_{p,w}$, $A_p \otimes A_{p,w}$, and $A_p \otimes A_p \otimes A_{p,w}$ are stable, then the forward recursions (6.98) and (6.99) are numerically stable and
the forward recursion (6.100) will converge to its steady-state solution.
Noticing that the eigenvalues of \( A_p \otimes A_{p,w} \) are the product of the eigenvalues of \( A_p \) and \( A_{p,w} \), a local window can allow the use of unstable systems (i.e., \( A_p \) unstable) while still having stable recursions. Among others, constant signals, polynomials and undamped sinusoids can be safely used with a proper window.

The analysis is analogous for the backward recursions (6.105)–(6.107) where stability of \( A_{f,w} \), \( A_f \otimes A_{f,w} \), and \( A_f \otimes A_f \otimes A_{f,w} \) is desired.

6.3 LSSM-Weighted Time-Dependent Polynomial Cost

Up until now, the cost functions that we have introduced in (6.1) and (6.87) are time-weighted squared error costs. In some cases, the squared error is not necessarily the discrepancy measure that we want to use. For this reason, we replace (and generalize) the sample-wise squared error by a sample-wise polynomial cost while still preserving a recursive cost computation. Parts of this section are covered in [89].

6.3.1 Cost Definition

Given a single-channel discrete-time signal \( y_1, \ldots, y_K \in \mathbb{R} \), where each sample \( y_i \) is associated with its own polynomial cost \( P_i \), and a LSSM signal \( f(C) \) with parameters \( \{C, A, s\} \) with unknown \( C = [C] \) (see (3.2)), let \( J_k(C) \) be the LSSM-weighted time-dependent polynomial cost

\[
J_k(C) = \sum_{i=1}^{K} w_{i-k} P_i (y_i - f_{i-k}(C)),
\]

where \( w_j, j \in \mathbb{Z} \), is a LSSM signal with fixed parameters \( \{C_w, A_w, s_w\} \). In general, all the polynomials \( P_i \) are chosen such that \( P_i(u) \geq 0 \), for all \( u \in \mathbb{R} \), but this is actually not a restriction. An important special case of (6.129) consists of polynomials \( P_i \) independent of \( i \) (i.e., \( P_i = P \), for all \( i \)), which leads to

\[
J_k(C) = \sum_{i=1}^{K} w_{i-k} P (y_i - f_{i-k}(C)).
\]

When \( P(u) = u^2 \), the cost (6.130) becomes the one of (6.87) (in the single-channel case).
The generalization to the case of $M$ uncorrelated channels leads to

$$J_k(C) = \sum_{m=1}^{M} J_k^{(m)}(C_m), \quad (6.131)$$

where each channel $m \in \{1, \ldots, M\}$ has its own cost $J_k^{(m)}(C_m)$ as in (6.129) depending on the $m^{th}$ row of $C$. This opens possibilities of having different LSSM windows and polynomial costs for each channel. A generalization to $M$ correlated channels is also possible by replacing the polynomial $P_i$ in (6.129) by a $M$-variate polynomial.

In the following, we focus on the single-channel case in order to keep the notation concise.

### 6.3.2 Recursive Computation of the Cost Function

Let $d \in \mathbb{N}$ denote the maximum degree of the polynomials $P_i, \ i \in \{1, \ldots, K\}$. We denote $p_j^{(i)}, \ j \in \{0, \ldots, d\}$ the coefficients of the polynomial $P_i$. The cost (6.129) can be recursively computed. Indeed, we have

$$J_k(C) = \sum_{i=1}^{K} w_{i-k} \sum_{j=0}^{d} \sum_{q=0}^{j} \frac{j}{q} (-1)^q \sum_{i=1}^{K} p_j^{(i)} y_i^{j-q} w_{i-k}(f_{i-k}(C))^q, \quad (6.132)$$

$$= \sum_{q=0}^{d} (-1)^q \sum_{i=1}^{K} \sum_{j=q}^{d} \frac{j}{q} p_j^{(i)} y_i^{j-q} w_{i-k}(f_{i-k}(C))^q, \quad (6.133)$$

$$= \sum_{q=0}^{d} (-1)^q \sum_{i=1}^{K} \tilde{g}_i^{(q)} w_{i-k}(f_{i-k}(C))^q, \quad (6.134)$$

with $\tilde{g}_i^{(q)} \in \mathbb{R}^K$ such that

$$\tilde{g}_i^{(q)} = \sum_{j=q}^{d} \frac{j}{q} p_j^{(i)} y_i^{j-q}, \quad (6.135)$$

for $i \in \{1, \ldots, K\}, \ q \in \{0, \ldots, d\}$.

Since for any $q \in \{0, \ldots, d\}, \ w_{i-k}(f_{i-k}(C))^q, \ i \in \mathbb{Z}$, is a LSSM signal shifted by a time $k$ and with LSSM parameters

$$\{(\otimes^q C) \otimes C_w, (\otimes^q A) \otimes A_w, (\otimes^q s) \otimes s_w\}, \quad (6.136)$$
Figure 6.1 – Graphical representation of the cost computation according to (6.143) and using (6.136), (6.140), and (3.104).

According to Proposition 4, we obtain that

$$
\sum_{i=1}^{K} \tilde{y}_i^{(q)} w_{i-k}(f_{i-k}(C))^q = \langle \tilde{y}^{(q)}, w_{i-k}(f_{i-k}(C))^q \rangle = [(\otimes^q C) \otimes C_w] \xi_k^{(q)},
$$

(6.138)

with, for $q \in \{0, \ldots, d\}$,

$$
\xi_k^{(q)} = \xi_k(\tilde{y}^{(q)}, (\otimes^q A) \otimes A_w, (\otimes^q s) \otimes s_w),
$$

(6.140)

using Proposition 9 and the function definition in (3.104). We use the convention that

$$
\otimes^q C = \underbrace{C \otimes \cdots \otimes C}_{q \text{ times}},
$$

(6.141)

for $q > 1$ and $\otimes^0 C = 1$ ( = $\{1, 1\}$). Note also that following our previous definitions and conventions, we have

$$
[(\otimes^q C) \otimes C_w] = [(\otimes^q C_p) \otimes C_{p,w} (\otimes^q C_f) \otimes C_{f,w}].
$$

(6.142)

Thus, the cost (6.129) is recursively and efficiently computed thanks to the relation

$$
J_k(C) = \sum_{q=0}^{d} (-1)^q [(\otimes^q C) \otimes C_w] \xi_k^{(q)}.
$$

(6.143)

For each $q \in \{0, \ldots, d\}$, the quantity $\xi_k^{(q)}$, defined in (6.140) with (3.104), is a linear function of $\tilde{y}^{(q)}$ but no longer of $y$. Furthermore, all the quantities $\xi_k^{(q)}$ can be efficiently computed using the forward and backward
recursions in (3.107) and (3.108). A graphical representation of formula (6.143) is given in Fig. 6.1.

The formula (6.143) also proves that \( \{ \xi_k^{(q)} : q \in \{0, \ldots, d\} \} \) is a finite-dimensional sufficient statistic for \( C \). The complexity of computing \( J_k \), for all \( k \in \{1, \ldots, K\} \), is only of \( O(K \max(n_p^{2d}n_{p,w}^{2d}, n_f^{2d}n_{f,w}^{2d})) \), which basically corresponds to the complexity of computing \( \xi_k^{(d)} \). In particular, whether the polynomials \( P_i \) are time-dependent or not, the computational complexity remains of the same order. Note also the squared dependency of the complexity with respect to the order of the LSSM window.

Since (6.1) and (6.87) are special cases of (6.129), the recursive computation implied by (6.143) generalizes the recursions derived in Sec. 6.1 and 6.2. In particular, in the single-channel case (i.e., \( M = 1 \)), we have the relations

\[
\xi_k(y^2, A_w, s_w) = \begin{bmatrix} \chi_k^+ \\ \chi_k^- \end{bmatrix} \tag{6.144}
\]

\[
\xi_k(y, A \otimes A_w, s \otimes s_w) = 2 \begin{bmatrix} \zeta_k^+ \\ \zeta_k^- \end{bmatrix} \tag{6.145}
\]

\[
\xi_k(y^0, (\otimes^2 A) \otimes A_w, (\otimes^2 s) \otimes s_w) = \begin{bmatrix} \text{vec}(\hat{S}_k^+) \\ \text{vec}(\hat{S}_k^-) \end{bmatrix}, \tag{6.146}
\]

referring to the quantities (6.95)–(6.97) and (6.102)–(6.104) and where \( y^j \) denotes the signal \( y \) raised element-wise to the power of \( j \in \mathbb{N}_0 \). While the proofs of (6.144) and (6.145) are straightforward, the proof of (6.146) requires formula (A.2).

### 6.3.3 Cost Function Minimization

The cost function (6.143) is a multivariate polynomial in \( C \in \mathbb{R}^{1 \times n} \). Thus, its minimization can be done using exact algebraic methods such as Gröbner bases or using a relaxation method such as a sum-of-squares (SOS) formulation solved by semidefinite programming (SDP) [57,67].
Chapter 7

Event Detection in Various Conditions

Estimating the parameters of an event is first conditioned on detecting the occurrence of an event. An event produces signals of interest that are often buried in noise or altered by some interference signal, which makes its detection not trivial and sometimes hard. After having defined a local likelihood [12], we propose to detect events using an appropriate hypothesis test with its associated (local) likelihood ratio.

Note that the local likelihood we define is based on a weighted squared error cost $J_k(C)$ as in (6.87) with positive window weights. The extension to a weighted time-dependent polynomial cost as in (6.129) is not straightforward and is not addressed here. However, the formulas for the various likelihood ratios usually depend on the cost function only and might be of practical use when naively using a cost as in (6.129).

7.1 Principles of Detection

7.1.1 Local Likelihood

Let $J_k(C)$ be the LSSM-weighted squared error cost (6.87) with positive window weights. The first attempt to define a likelihood function for the observations $y_1, \ldots, y_K$ given the presence at time index $k$ of a signal
parametrized by $C$ leads to

$$p(y_1, \ldots, y_K | C, \sigma_Z^2) \propto \exp \left( - \frac{J_k(C)}{2 \sigma_Z^2} \right), \quad (7.1)$$

for some $\sigma_Z^2 \in \mathbb{R}_+$ and where $\propto$ means equality up to a scale factor independent of $C$. In this way, maximizing the likelihood for a given $\sigma_Z^2$ is equivalent to minimizing $J_k(C)$. Incorporating the missing scale factor in (7.1), we have

$$p(y_1, \ldots, y_K | C, \sigma_Z^2) = \sqrt{\frac{\prod_{i=1}^{K} (w_{i-k})^M}{(2\pi \sigma_Z^2)^{KM}}} \exp \left( - \frac{J_k(C)}{2 \sigma_Z^2} \right) \quad (7.2)$$

$$= \prod_{i=1}^{K} \mathcal{N} \left( y_i : f_{i-k}(C), \frac{\sigma_Z^2}{w_{i-k}} I_M \right), \quad (7.3)$$

which is equivalent to the statistical assumption

$$\sqrt{w_{i-k}} (y_i - f_{i-k}(C)) \overset{iid}{\sim} \mathcal{N}(0, \sigma_Z^2 I_M). \quad (7.4)$$

Thus, the contribution of sample $y_i$ in the likelihood (7.2) creates a factor

$$\mathcal{N} \left( y_i : f_{i-k}(C), \frac{\sigma_Z^2}{w_{i-k}} I_M \right), \quad (7.5)$$

which typically goes to zero as $|i - k| \to +\infty$ since $w_{i-k}$ goes to zero (in the case of a finite window it even becomes exactly zero). Thus, those factors badly influence the likelihood (7.2) which is supposed to locally measure the goodness of fit around time index $k$. In particular, the noise variance maximizing (7.2)

$$\hat{\sigma}_Z^2 = \frac{1}{KM} J_k(C) = \frac{1}{KM} \sum_{i=1}^{K} w_{i-k} \left\| y_i - f_{i-k}(C) \right\|^2 \quad (7.6)$$

is a nonsensical estimate due to an inappropriate scaling with respect to the number of samples.

To circumvent those issues, we define the local likelihood

$$\tilde{p}(y_1, \ldots, y_K; C, \sigma_Z^2) = \prod_{i=1}^{K} \mathcal{N} \left( y_i : f_{i-k}(C), \frac{\sigma_Z^2}{w_{i-k}} I_M \right)^{w_{i-k}} \quad (7.7)$$

$$= \frac{1}{(2\pi \sigma_Z^2)^{\frac{v_{k.M}}{2}}} \exp \left( - \frac{J_k(C)}{2 \sigma_Z^2} \right), \quad (7.8)$$

where $v_{k.M} = \sum_{i=1}^{K} w_{i-k}$.
with the window area \( \nu_k = \sum_{i=1}^{K} w_{i-k} \).

The local likelihood does not integrate to one. Nonetheless, \( C \) that minimizes \( J_k(C) \) also maximizes the local likelihood. Furthermore, the contribution of sample \( y_i \) in the likelihood (7.2) creates a factor

\[
\mathcal{N}(y_i : f_{i-k}(C), \sigma_Z^2 I_M)^{w_{i-k}},
\]

which typically goes to one as \( |i-k| \to +\infty \) (since \( w_{i-k} \) goes to zero). Thus, the factors in (7.7) corresponding to samples far from index \( k \) barely influence the local likelihood. In particular, the noise variance which maximizes the local likelihood

\[
\hat{\sigma}_Z^2 = \frac{1}{\nu_k M} J_k(C) = \frac{1}{M} \sum_{i=1}^{K} w_{i-k} \left\| y_i - f_{i-k}(C) \right\|^2 / \sum_{i=1}^{K} w_{i-k}
\]

is now meaningful and corresponds to the weighted per-sample per-channel squared error. Instead of using the standard likelihood, we thus use the local likelihood.

Denoting \( e = \exp(1) \), we also have

\[
\max_{\sigma_Z^2} \ln \hat{p}(y_1, \ldots, y_K; C, \sigma_Z^2) = -\frac{\nu_k M}{2} \ln \left( 2\pi e \frac{J_k(C)}{\nu_k M} \right).
\]

\[
(7.11)
\]

### 7.1.2 Detection as Hypothesis Test

In order to distinguish between LSSM signals from a given set \( C_1 \) and signals from \( C_0 \) at time index \( k \), we use the following hypothesis test

- \( H_0: C \in C_0 \)
- \( H_1: C \in C_1 \)

with the per-sample per-channel log-likelihood ratio

\[
\text{LLR}_k = \frac{1}{\nu_k M} \ln \left( \frac{\max_{C \in C_1, \sigma_Z^2} \hat{p}(y_1, \ldots, y_K; C, \sigma_Z^2)}{\max_{C \in C_0, \sigma_Z^2} \hat{p}(y_1, \ldots, y_K; C, \sigma_Z^2)} \right)
\]

\[
= -\frac{1}{2} \ln \left( \frac{\min_{C \in C_1} J_k(C)}{\min_{C \in C_0} J_k(C)} \right),
\]

\[
(7.13)
\]

where we have used (7.11). If \( \text{LLR}_k \) is above a given threshold and locally maximum then a signal from \( C_1 \) is detected. We now emphasize specific detection situations.
Signal Versus Noise

If a signal from a given set $C$ is buried in white noise we can use $C_1 = C$ and $C_0 = \{0\}$ (noise-only hypothesis). This test will be successful even in the presence of additive interference signals that are almost not correlated with the signals from $C$.

Signal Versus Anything

The detection of a signal from a given set $C$ might be done using $C_1 = C$ and $C_0 = \mathbb{R}^{M \times n}$. In particular, this test outputs good detection abilities when $C$ is a finite set.

7.2 Detection in the Presence of Interferences

When an event occurs, the characteristic LSSM signal that is produced may be altered by some interference signal. Those interferences need to be taken into account in both the detection and the estimation problems. A standard approach would be to filter out interference signals in a preprocessing step. Instead, we model a multi-channel interference signal with its own LSSM signal $g(C_g)$ with parameters $\{C_g, A_g, s_g\}$ but unknown $C_g \in C_g$. The task is still to detect and estimate a LSSM signal $f(C)$ of interest with parameters $\{C, A, s\}$ with unknown $C \in C$. In the following, we deal with additive and multiplicative interferences.

7.2.1 Additive Interference

In order to deal with an additive interference, we introduce the cost

$$J_k(C, C_g) = \sum_{i=1}^{K} w_{i-k} \| y_i - (g_{i-k}(C_g) + f_{i-k}(C)) \|_2^2,$$  

(7.14)

based on a LSSM-weighted squared error cost as in (6.87). The generalization to the LSSM-weighted time-dependent polynomial cost is straightforward.

Since $f(C) + g(C_g)$ is also a LSSM signal (see Proposition 5) with left-sided parameters

$$\left\{ \begin{bmatrix} C_p & C_{p,g} \end{bmatrix}, \text{diag}(A_p, A_{p,g}), \begin{bmatrix} s_p \end{bmatrix}, \begin{bmatrix} s_{p,g} \end{bmatrix} \right\}$$  

(7.15)
and right-sided parameters

\[
\begin{bmatrix}
C_f & C_{f,g} \\
\end{bmatrix}, \text{diag}(A_f, A_{f,g}), \begin{bmatrix} s_f \\ s_{f,g} \end{bmatrix}
\]\n
(i.e., a proper stacking of \(\{C_g, A_g, s_g\}\) with \(\{C, A, s\}\)), \(J_k(C, C_g)\) can be recursively computed using the same recursions as in Sec. 6.2.2 but using these stacked parameters. Furthermore, the minimization problem

\[
(\hat{C}, \hat{C}_g) = \arg\min_{C \in \mathcal{C}, C_g \in \mathcal{C}_g} J_k(C, C_g)
\]

is of the same nature as (6.30). Based on this cost, we now propose two likelihood ratio tests for detecting LSSM signals from a given set \(\mathcal{C}\) occurring on top of an interference signal.

**Signal Plus Interference Versus Interference**

For such a detection task, a meaningful hypothesis test is

- \(H_0: C = 0, C_g \in \mathcal{C}_g\)
- \(H_1: C \in \mathcal{C}, C_g \in \mathcal{C}_g\)

with its per-sample per-channel log-likelihood ratio (cf. (7.13))

\[
\text{LLR}_k = -\frac{1}{2} \ln \left( \frac{\min_{C \in \mathcal{C}, C_g \in \mathcal{C}_g} J_k(C, C_g)}{\min_{C_g \in \mathcal{C}_g} J_k(C = 0, C_g)} \right).
\]

However, as can be seen from this ratio, good detection abilities require that the interference model alone cannot explain the signals of interest well enough. Otherwise, the denominator will often be of similar magnitude as the numerator in (7.18).

In order to avoid the trade-off between having a flexible interference model and achieving a good detection performance, we propose another hypothesis test.

**Given Interference Estimated from Joint Model, Signal or Interference Only**

Let \((\hat{C}, \hat{C}_g)\) be as defined in (7.17). Given the interference signal estimated from the joint model (signal plus interference), we test whether it
Event Detection in Various Conditions

is more likely to have a signal plus interference or an interference signal only with the quantity

$$\text{LLR}_k = -\frac{1}{2} \ln \left( \frac{\min_{C \in \mathcal{C}, C_g \in \mathcal{C}_g} J_k(C, C_g)}{J_k(C = 0, \hat{C}_g)} \right). \tag{7.19}$$

Compared to (7.18), we here use the same interference signal estimated from the joint LSSM in both hypotheses. This ratio measures the gain of using the signal model part within the joint model of signal and interference. Unlike (7.18), it does not suffer from having a flexible interference model as long as the interference model does not completely supersede the signal model.

### 7.2.2 Multiplicative Interference

In order to deal with a multiplicative interference, we introduce the cost

$$J_k(C, C_g) = \sum_{i=1}^{K} w_{i-k} \| y_i - g_{i-k}(C_g) \cdot f_{i-k}(C) \|^2, \tag{7.20}$$

based on a LSSM-weighted squared error cost as in (6.87). The product sign “·” in (7.20) is to be understood as element-wise product between elements of two vectors. The generalization to the LSSM-weighted time-dependent polynomial cost is straightforward.

Since $f(C) \cdot g(C_g)$ is also a multi-channel LSSM signal (see Proposition 4) with left-sided parameters

$$\{ C_p \tilde{\otimes} C_{p,g}, A_p \otimes A_{p,g}, s_p \otimes s_{p,g} \} \tag{7.21}$$

and right-sided parameters

$$\{ C_f \tilde{\otimes} C_{f,g}, A_f \otimes A_{f,g}, s_f \otimes s_{f,g} \} \tag{7.22}$$

(where $\tilde{\otimes}$ denotes the Kronecker product applied separately for each row, when both arguments have the same number of rows), $J_k(C, C_g)$ can be recursively computed using the same recursions as in Sec. 6.2.2 but using these modified parameters. However, the minimization problem

$$(\hat{C}, \hat{C}_g) = \arg \min_{C \in \mathcal{C}, C_g \in \mathcal{C}_g} J_k(C, C_g) \tag{7.23}$$

is not of the form (6.30) but consists in minimizing a multivariate polynomial, which can be done as in Sec. 6.3.3 As can be seen in (7.21)
and (7.22), there is an undetermined scale factor between each row of \( C_p \) and \( C_{p,g} \) and also between each row of \( C_f \) and \( C_{f,g} \). Depending on \( C \) and \( C_g \), further constraints can be added to get rid of this ambiguity.

Based on this cost, we now propose two likelihood ratio tests for detecting LSSM signals from a given set \( C \) occurring multiplicatively with an interference signal.

**Signal With Multiplicative Interference Versus Noise**

If interference signals occur only when signals of interest are present (on/off scenario), we can use the hypothesis test

- \( H_0: C = 0, C_g = 0 \)
- \( H_1: C \in C, C_g \in C_g \)

with its per-sample per-channel log-likelihood ratio (cf. (7.13))

\[
\text{LLR}_k = -\frac{1}{2} \ln \left( \frac{\min_{C \in C, C_g \in C_g} J_k(C, C_g)}{J_k(C = 0, C_g = 0)} \right). \tag{7.24}
\]

Note that when \( C = 0 \), the interference signal has anyway no effect on the cost function.

**Signal With Multiplicative Interference Versus Interference**

If interference signals are present whether or not signals of interest occur, we can use the hypothesis test

- \( H_0: C_g \in C_g \)
- \( H_1: C \in C, C_g \in C_g \)

with its per-sample per-channel log-likelihood ratio (cf. (7.13))

\[
\text{LLR}_k = -\frac{1}{2} \ln \left( \frac{\min_{C \in C, C_g \in C_g} J_k(C, C_g)}{J_k^{(g)}(C_g)} \right), \tag{7.25}
\]

where \( J_k^{(g)}(C_g) \) denotes a cost as in (7.20) with \( f = 1 \) (i.e., the constant unit signal).
7.3 Detection and Classification

Assume that we want to detect and classify LSSM signals from $L$ different classes denoted by $C_\ell$, $\ell \in \{1, \ldots, L\}$. Assume also that we have a null class $C_0$, which contains events to be ignored.

7.3.1 Using Multiple Hypothesis Tests

A natural approach for detecting and classifying LSSM signals from $L$ different classes is to define $L$ hypothesis tests

- $H_0$: $C \in C_0$
- $H_\ell$: $C \in C_\ell$

with the per-sample per-channel log-likelihood ratio (cf. (7.13))

$$\text{LLR}_k^{(\ell)} = - \frac{1}{2} \ln \left( \frac{\min_{C \in C_\ell} J_k(C)}{\min_{C \in C_0} J_k(C)} \right), \quad (7.26)$$

for $\ell \in \{1, \ldots, L\}$. Then, the quantity we use for detection is

$$\text{LLR}_k = \max_{\ell \in \{1, \ldots, L\}} \text{LLR}_k^{(\ell)}. \quad (7.27)$$

If $\text{LLR}_k$ is above a given threshold and locally maximum then a signal from $C_\hat{\ell}$ is detected, where

$$\hat{\ell} = \arg\max_{\ell \in \{1, \ldots, L\}} \text{LLR}_k^{(\ell)}. \quad (7.28)$$

7.3.2 Using a Classifier

In the multiple hypothesis test case, $L + 1$ minimization problems need to be solved at each time index $k$, which can be costly in some situations. A different approach consists in using the hypothesis test

- $H_0$: $C \in C_0$
- $H_1$: $C \in \bigcup_{\ell=1}^{L} C_\ell$

with its per-sample per-channel log-likelihood ratio (cf. (7.13)) $\text{LLR}_k$.

If $\text{LLR}_k$ is above a given threshold and locally maximum then a signal of interest is detected. Finally, we compute

$$\hat{C} = \arg\min_{C \in \bigcup_{\ell=1}^{L} C_\ell} J_k(C). \quad (7.29)$$
and use a classifier (i.e., a mapping $\mu : \mathbb{R}^{M \times n} \rightarrow \{1, \ldots, L\}$) to assign $\hat{C}$ to a class.

The classifier can be set a priori based on the knowledge of $C_\ell$, $\ell \in \{1, \ldots, L\}$, or can also be learned from data (in a supervised or unsupervised way).

### 7.4 Detection of Model Switches in Model-Dependent Noise

The LSSM-weighted squared error cost introduced in (6.87) is a sum of a cost $\hat{J}_k(C_p)$ for the “past” model and a cost $\hat{J}_k(C_f)$ for the “future” model. When introducing the local likelihood in (7.7), we have assumed that the LSSM-weighted average squared error $\sigma_Z^2$ is equally distributed among samples contributing in both the “past” and “future” models. Considering a distinct noise variance for each model opens new possibilities for detection.

#### 7.4.1 Two-Sided Local Likelihood

In order to deal with a model-dependent noise variance, we define a two-sided local likelihood

$$\tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2) = \tilde{p}(y_1, \ldots, y_k; C_p, \sigma_p^2) \tilde{p}(y_{k+1}, \ldots, y_K; C_f, \sigma_f^2),$$

with

$$\tilde{p}(y_1, \ldots, y_k; C_p, \sigma_p^2) = \prod_{i=1}^{k} \mathcal{N}(y_i : f_{i-k}(C), \sigma_p^2 I_M)^{w_{i-k}}$$

$$\quad = \frac{1}{(2\pi\sigma_p^2)^{\frac{kM}{2}}} \exp \left( -\frac{\hat{J}_k(C_p)}{2\sigma_p^2} \right),$$

with the window area $\tilde{\nu}_k = \sum_{i=1}^{k} w_{i-k}$ and

$$\tilde{p}(y_{k+1}, \ldots, y_K; C_f, \sigma_f^2) = \prod_{i=k+1}^{K} \mathcal{N}(y_i : f_{i-k}(C), \sigma_f^2 I_M)^{w_{i-k}}$$

$$\quad = \frac{1}{(2\pi\sigma_f^2)^{\frac{kM}{2}}} \exp \left( -\frac{\hat{J}_k(C_f)}{2\sigma_f^2} \right),$$
with the window area \( \mathcal{V}_k = \sum_{i=k+1}^{K} w_{i-k} \). Thus, we also have
\[
\tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2) = \frac{1}{((2\pi)^{\nu_k}(\sigma_p^2)^{\nu_k}(\sigma_f^2)^{\nu_k})^M} \exp \left( -\frac{1}{2} \left( \frac{\tilde{J}_k(C_p)}{\sigma_p^2} + \frac{\tilde{J}_k(C_f)}{\sigma_f^2} \right) \right). \tag{7.35}
\]

In particular, the noise variances which maximize the two-sided local likelihood
\[
\hat{\sigma}^2_p = \frac{1}{\nu_k M} \tilde{J}_k(C_p) \tag{7.36}
\]
\[
\hat{\sigma}^2_f = \frac{1}{\nu_k M} \tilde{J}_k(C_f) \tag{7.37}
\]
are now different for each model side.

Note also that
\[
\max_{\sigma_p^2, \sigma_f^2} \frac{1}{\nu_k M} \ln \tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2) = -\frac{1}{2} \left( \ln \left( \frac{2\pi e}{M} \right) + \frac{\nu_k}{\nu_k} \ln \left( \frac{\tilde{J}_k(C_p)}{\nu_k} \right) + \frac{\nu_k}{\nu_k} \ln \left( \frac{\tilde{J}_k(C_f)}{\nu_k} \right) \right). \tag{7.38}
\]

### 7.4.2 Another Family of Hypothesis Tests

Based on the two-sided local likelihood, a new hypothesis test can be expressed using the model-dependent variances. In addition to distinguishing between two LSSM signal sets \( C_0 \) and \( C_1 \), we further restrict \((\sigma_p^2, \sigma_f^2)\) to belong to \( S_0 \subset \mathbb{R}^2_+ \) when a LSSM signal from \( C_0 \) occurs and to \( S_1 \subset \mathbb{R}^2_+ \) when a LSSM signal from \( C_1 \) occurs. Thus, we define the test

- \( \mathcal{H}_0: C \in C_0, (\sigma_p^2, \sigma_f^2) \in S_0 \)
- \( \mathcal{H}_1: C \in C_1, (\sigma_p^2, \sigma_f^2) \in S_1 \)

with the per-sample per-channel log-likelihood ratio
\[
\text{LLR}_k = \frac{1}{\nu_k M} \ln \left( \frac{\max_{C \in \mathcal{C}_0, (\sigma_p^2, \sigma_f^2) \in S_0} \tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2)}{\max_{C \in \mathcal{C}_1, (\sigma_p^2, \sigma_f^2) \in S_1} \tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2)} \right). \tag{7.39}
\]
7.4.3 Detection of Model Switches

We use the term model switches for particular sets of events consisting of a change from a “past” model to a “future” one but for which there is no link between the “past” and the “future” signal. Specifically, upon occurrence of a model switch, \( C_p \) and \( C_f \) can be optimized separately, that is to say, there are only individual restrictions \( C_p \in C_p \) and \( C_f \in C_f \).

In addition, when a switch occurs, a change of noise variance can happen (i.e., \( \sigma_p \neq \sigma_f \)) whereas when no switch occurs \( C = [C_p \quad C_f] \in C_0 \) and the noise variance stays the same (i.e., \( \sigma_p^2 = \sigma_f^2 \)).

In order to detect model switches, we can use a hypothesis test as described in Sec. 7.4.2 with

- \( H_0: [C_p \quad C_f] \in C_0, \sigma_p^2 = \sigma_f^2 \)
- \( H_1: C_p \in C_p, \sigma_p^2 \in \mathbb{R}^+, C_f \in C_f, \sigma_f^2 \in \mathbb{R}^+ \),

with the per-sample per-channel log-likelihood ratio

\[
LLR_k = \frac{1}{\nu_k M} \ln \left( \frac{\max_{\underbrace{H_1}} \tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2)}{\max_{\underbrace{H_0}} \tilde{p}(y_1, \ldots, y_K; C_p, \sigma_p^2, C_f, \sigma_f^2)} \right). \tag{7.40}
\]

Under \( H_0 \), the two-sided local likelihood becomes the local likelihood in (7.7) and thus (7.11) can be used for the maximization of the denominator in (7.40). For the numerator, (7.38) can be used. Thus, we obtain

\[
LLR_k = -\frac{1}{2} \ln \left( \left( \min_{C_p \in C_p} \frac{\tilde{J}_k(C_p)}{\nu_k} \right) \left( \min_{C_f \in C_f} \frac{\tilde{J}_k(C_f)}{\nu_k} \right) \left( \min_{C \in C_0} \frac{\tilde{J}_k(C)}{\nu_k} \right) \right). \tag{7.41}
\]

All minimization problems in (7.41) consist of a problem of the form (6.30).
In order to illustrate the capabilities of the approach proposed in Part II, we apply it to several practical applications. In all those examples, signals produced by events of interest are modeled with two-sided LSSM signals. Then, a local cost among the ones introduced in Chapter 6 is used and computed efficiently thanks to recursive expressions. Finally, a suitable hypothesis test with its associated likelihood ratio, as suggested in Chapter 7, is used to detect events and estimate parameters of such events.

While computing a cost function as defined in Chapter 6, both forward and backward recursions are required unless we have a left-sided LSSM only. Even if the forward recursions have a purely online implementation, the backward recursions unfortunately prevent our detection and estimation algorithm from having an online implementation. Indeed, for detecting an event at time index $k$, all samples from $k$ up to the end are needed for the backward recursions. However, by computing the backward recursions on a finite window (which is often a good enough approximation), we slightly modify our algorithm such that an online implementation with the use of a buffer becomes feasible. Therefore, our original offline algorithm can be readily modified into a practical online algorithm, which opens possibilities for real-time implementations.
8.1 Robust ECG Shape Detection and Separation

We consider the well-known problem of heart beat detection in a single-channel surface ECG recording (lead II), as plotted in Fig. 8.1. Looking closer at one ECG peak in Fig. 8.1, we observe an ECG signal shape as in Fig. 8.2. The goal is to robustly detect each repetition of such ECG signal shape even in the presence of a wandering baseline. We do not claim that our approach outperforms existing methods in this case (although it works very well). Our main point here is to illustrate the proposed approach with an example that is easier to understand than the more complex examples of Sec. 8.2 and 8.3.

8.1.1 LSSM for Measurement Signals

LSSM for the ECG Shape

We model an ECG signal shape as in Fig. 8.2 with a two-sided LSSM. The left-sided model of order $n_p^D = 8$ is parametrized by $C_p^D$, $A_p^D$, and $s_p^D$ and consists of a superposition of 4 exponentially decaying sinusoids.
The right-sided model of order \( n_D = 16 \) is parametrized by \( C_D, A_D, \) and \( s_D \) and consists of a superposition of 8 exponentially decaying sinusoids. In Fig. 8.2 we plot the ECG signal shape and its LSSM signal.

**LSSM for Interference Signals**

In practical applications, the ECG shapes are affected by external disturbances such as the patient’s breathing or body movements, which create a wandering baseline as in Fig. 8.1. Selecting a pre-processing filter that does not distort the ECG shapes is quite cumbersome. Instead we model additive interferences with a discrete-time polynomial of degree \( n_I - 1 \) that has a LSSM of order \( n_I \) and parameters \( A_I = I_{n_I} + N_{n_I} \in \mathbb{R}^{n_I \times n_I}, \) \( s_I = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix}^T \in \mathbb{R}^{n_I}, \) and observation vector \( C_I \in \mathbb{R}^{1 \times n_I} \) depending on the coefficients of the polynomial (cf. Sec. 3.3).

**Joint LSSM for the Measurements**

The joint LSSM parameters are

\[
\begin{align*}
C_p &= \begin{bmatrix} \lambda C_p^D & C_1 \end{bmatrix} \in \mathbb{R}^{1 \times (n_p + n_I)} \\
A_p &= \text{diag}(A_p^D, A_I^{-1}) \in \mathbb{R}^{(n_p + n_I) \times (n_p + n_I)} \\
s_p &= \begin{bmatrix} (s_p^D)^T & s_I^T \end{bmatrix}^T \in \mathbb{R}^{(n_p + n_I)} \\
C_f &= \begin{bmatrix} \lambda C_f^D & C_1 \end{bmatrix} \in \mathbb{R}^{1 \times (n_f + n_I)} \\
A_f &= \text{diag}(A_f^D, A_I) \in \mathbb{R}^{(n_f + n_I) \times (n_f + n_I)} \\
s_f &= \begin{bmatrix} (s_f^D)^T & s_I^T \end{bmatrix}^T \in \mathbb{R}^{(n_f + n_I)}.
\end{align*}
\]
Note that an unknown parameter $\lambda \in \mathbb{R}_+$ has been introduced in our model to cope with amplitude scales of the ECG signal shape. The observation vector $C = [C_p \ C_f]$ decomposes into

$$C = \lambda C_D P_D + C_I P_I,$$

(8.7)

with $C_D = [C_D^p \ C_D^f]$, 

$$P_D = \begin{bmatrix} I_{n_p^D} & 0_{n_p^D \times n_I^D} & 0_{n_f^D \times n_I^D} & 0_{n_p^D \times n_I^f} \\ 0_{n_f^D \times n_p^D} & I_{n_I^D} & 0_{n_f^D \times n_I^f} & 0_{n_I^D} \end{bmatrix}$$

(8.8)

$$P_I = \begin{bmatrix} 0_{n_I \times n_p^D} & I_{n_I} & 0_{n_I \times n_f^D} & I_{n_I} \end{bmatrix}.$$

(8.9)

### 8.1.2 Detecting ECG Signal Shapes

Given an ECG signal $y_1, \ldots, y_K \in \mathbb{R}$ as in Fig. 8.1, we want to detect the ECG signal shape of Fig. 8.2. Since our algorithm incorporates a polynomial model for the interference signal, we use the local cost $J_k(C)$ as in (6.1) with an exponential window with $(\gamma_p, \gamma_f) \in (0, 1)^2$.

#### Recursive Cost Computation

The cost $J_k(C)$ is efficiently computed using forward/backward recursions as described in Sec. 6.1.2 and expresses as (6.22).

#### Minimizing the Cost Function

We focus on minimizing $J_k$ with respect to $C$ as in (8.7) with $\lambda \in \mathbb{R}_+$ and $C_I \in \mathbb{R}^{1 \times n_I}$. The minimization over $C_I$ is a “Constrained Offset” problem of Table 6.2. Thus, we have

$$\hat{C}_I = (\xi_k^T - \lambda C_D P_D W_k) P_I (P_I W_k P_I^T)^{-1}$$

(8.10)

and after simplifications, using “Post-Multiplied” of Table 6.2,

$$\min_{C_I \in \mathbb{R}^{1 \times n_I}} J_k(\lambda, C_I) = \tilde{\kappa}_k - 2\lambda \tilde{\xi}_k + \lambda^2 \tilde{W}_k,$$

(8.11)

with

$$\tilde{\kappa}_k = \kappa_k - \text{tr}(\xi_k^T P_I (P_I W_k P_I^T)^{-1} P_I \xi_k)$$

(8.12)

$$\tilde{\xi}_k = C_D P_D (\xi_k - W_k P_I (P_I W_k P_I^T)^{-1} P_I \xi_k)$$

(8.13)

$$\tilde{W}_k = C_D P_D (W_k - W_k P_I (P_I W_k P_I^T)^{-1} P_I W_k) P_I^T C_D^T.$$
Then, the remaining minimization over $\lambda \in \mathbb{R}_+$ leads to
\[
\hat{\lambda} = \max(\bar{\xi}_k / \bar{W}_k, 0)
\] (8.15)
and
\[
\min_{\lambda \in \mathbb{R}_+, C_1 \in \mathbb{R}^{1 \times n_1}} J_k(\lambda, C_1) = \begin{cases} 
\bar{\kappa}_k, & \text{for } \bar{\xi}_k < 0 \\
\bar{\kappa}_k - \frac{\bar{\xi}_k^2}{W_k}, & \text{for } \bar{\xi}_k \geq 0.
\end{cases}
\] (8.16)

**Detection and Estimation**

In order to detect an ECG signal shape at time index $k$, given the polynomial coefficients $\hat{C}_1$ estimated with the joint LSSM, we use the hypothesis test

- $\mathcal{H}_0$: $\lambda = 0$
- $\mathcal{H}_1$: $\lambda > 0$

and the following log-likelihood ratio (cf. Sec. 7.2.1)
\[
\text{LLR}_k = -\frac{1}{2} \ln \left( \frac{\min_{\lambda \in \mathbb{R}_+} J_k(\lambda, \hat{C}_1)}{J_k(\lambda = 0, \hat{C}_1)} \right).
\] (8.17)

If $\text{LLR}_k$ is above a given threshold and locally maximum then an ECG signal shape is detected. In Fig. 8.1, we plot an ECG recording with its associated signal $\text{LLR}_k$. The vertical red dashed lines indicate the peaks of $\text{LLR}_k$, which coincide with the presence of an actual ECG signal shape. Even in the presence of an unknown wandering baseline, our algorithm outputs a robust detection and estimation of the ECG signal shapes.

### 8.2 Detecting and Estimating Straight-Line Movements of a Magnet From Magnetic Field Measurements

Contemporary smartphones embed a magnetometer that measures the magnetic field in 3D. Sweeping over the phone with a magnet (see illustrative plot in Fig 8.3) induces magnetic field variations which are
measured by the magnetometer. Given these 3-channel signals, as plotted in Fig. 8.5, we design an algorithm based on our approach to detect straight-line passes of a magnet and estimate their speed and direction. This part extends [84].

8.2.1 Physical Model of the Magnetic Field

We use the coordinate system of the magnetometer as canonical basis. Let \( \mathbf{r}_k \in \mathbb{R}^3 \) be the magnet position at time \( k \in \mathbb{Z} \). The magnetic field vector \( \mathbf{B}_k \in \mathbb{R}^3 \) induced by a dipole at position \( \mathbf{r}_k \) is

\[
\mathbf{B}_k = \frac{\mu_0}{4\pi} \left( \frac{3(\mathbf{m}, \mathbf{r}_k)\mathbf{r}_k}{\|\mathbf{r}_k\|^5} - \frac{\mathbf{m}}{\|\mathbf{r}_k\|^3} \right),
\]

where \( \mathbf{m} \in \mathbb{R}^3 \) is the magnetic moment of the magnet and \( \mu_0 = 4\pi \cdot 10^{-7} \text{ N/A}^2 \) is the vacuum permeability. Note that quantities in \( \mathbb{R}^3 \) are highlighted in bold.

We consider a discrete uniform rectilinear movement \( \mathbf{r}_k \in \mathbb{R}^3, k \in \mathbb{Z} \), which starts and ends far away from the magnetometer (i.e., \( \|\mathbf{r}_k\| \to +\infty \) as \( k \to \pm \infty \)). Without loss of generality we assume that the minimum distance \( r \) is reached at \( k = 0 \) and thus, such a movement can be
8.2 Detecting and Estimating Straight-Line Movements of a Magnet From Magnetic Field Measurements

parametrized as

\[ \mathbf{r}_k = r_d \mathbf{d}_r + kv \mathbf{d}_v = r (\mathbf{d}_r + \frac{k}{\tau} \mathbf{d}_v), \]  

(8.19)

where \( r \in \mathbb{R}^*_+ \) is the minimum distance to the magnetometer, \( \mathbf{d}_r \in S^3 \) (with \( S^3 \) the unit sphere in \( \mathbb{R}^3 \)) is the direction when achieving this minimum distance, \( v \in \mathbb{R}^*_+ \) is the constant per-sample speed, \( \mathbf{d}_v \in S^3 \) is the direction of the movement, and \( \tau = r/v \) is the time-scale parameter. Note that \( \langle \mathbf{d}_r, \mathbf{d}_v \rangle = 0 \) due to the minimum distance characterization of \( \mathbf{d}_r \) and thus \( \| \mathbf{r}_k \|^2 = r^2 (1 + \left( \frac{k}{\tau} \right)^2) \).

The moment \( \mathbf{m} = m \mathbf{d}_m \) (\( m \in \mathbb{R}^*_+, \mathbf{d}_m \in S^3 \)) is considered independent of time (i.e., no internal rotation of the magnet during a movement). We parametrize \( \mathbf{d}_m \) with respect to the movement basis \((\mathbf{d}_r, \mathbf{d}_v, \mathbf{d}_{rv})\) with \( \mathbf{d}_{rv} = \mathbf{d}_r \times \mathbf{d}_v \):

\[ \mathbf{d}_m = \langle \mathbf{d}_m, \mathbf{d}_r \rangle \mathbf{d}_r + \langle \mathbf{d}_m, \mathbf{d}_v \rangle \mathbf{d}_v + \langle \mathbf{d}_m, \mathbf{d}_{rv} \rangle \mathbf{d}_{rv}. \]  

(8.20)

Finally, plugging (8.19) and (8.20) into (8.18), we get

\[ \mathbf{B}_k = \lambda RFHG \left( \frac{k}{\tau} \right), \]  

(8.21)

with

\[ G(t) = \frac{1}{(1 + t^2)^{3/2}} \begin{bmatrix} 1 - \frac{1}{2} t^2 & t \\ t & 1 + t^2 \end{bmatrix} \]  

(8.22)

\[ H = \text{diag}(2, 3, -1) \]  

(8.23)

\[ F = \begin{bmatrix} \langle \mathbf{d}_m, \mathbf{d}_r \rangle & \langle \mathbf{d}_m, \mathbf{d}_v \rangle & 0 \\ -\langle \mathbf{d}_m, \mathbf{d}_v \rangle & \langle \mathbf{d}_m, \mathbf{d}_r \rangle & -\langle \mathbf{d}_m, \mathbf{d}_v \rangle \\ 0 & 0 & \langle \mathbf{d}_m, \mathbf{d}_{rv} \rangle \end{bmatrix} \]  

(8.24)

\[ R = \begin{bmatrix} \mathbf{d}_r & \mathbf{d}_v & \mathbf{d}_{rv} \end{bmatrix} \in SO_3(\mathbb{R}) \]  

(8.25)

\[ \lambda = \frac{\mu_0 m}{4\pi r^3} \in \mathbb{R}_+. \]  

(8.26)

The parameter \( \tau \) essentially accounts for the time scale (i.e., the signals are roughly supported on \([-8\tau, 8\tau]\), \( R \) contains the movement direction, and \( F \) accounts for the moment orientation with respect to the movement direction. As a movement starts and ends far away from the magnetometer, \( \mathbf{B}_k \to 0 \) as \( k \to \pm \infty \).

With this parametrization, we observe that both \((\lambda, \mathbf{d}_r, \mathbf{d}_v, \mathbf{d}_m, \tau)\) and \((\lambda, -\mathbf{d}_r, -\mathbf{d}_v, \mathbf{d}_m, \tau)\) generate the same discrete-time signal \( \mathbf{B}_k, k \in \mathbb{Z} \).
Z. In order to ensure identifiability, we need to restrict the area where the magnet can move or the way the magnet is oriented. For instance if the magnetometer is put on a surface, we can always ensure that \( \mathbf{d}_r \) has a positive \( z \)-coordinate with respect to the magnetometer axes. Alternatively, we can also orient the magnet such that \( \langle \mathbf{d}_m, \mathbf{d}_r \rangle > 0 \) while a magnet is moving, which makes the parameter set identifiable.

8.2.2 LSSM for Magnetic Field Signals

LSSM for the Magnetic Field of the Dipole

Looking at Fig. 8.4, it seems that modeling each component of \( \mathbf{G}(t) \) using a two-sided 2\(^{nd}\)-order LSSM is good enough for our gesture recognition purposes. Taking into consideration the symmetry properties and the scaling of the components of \( \mathbf{G}(t) \), the estimated 3D-signal has the form

\[
\hat{\mathbf{G}}(t) = \begin{bmatrix}
e^{-\alpha_x |t| \cos(\omega_x |t|+\Phi_x)} \\
K_y e^{-\alpha_y |t|} \sin(\omega_y t) \\
e^{-\alpha_z |t| \cos(\omega_z |t|+\Phi_z)}
\end{bmatrix}.
\]

The parameters \( \alpha_x, \omega_x, \Phi_x, K_y, \alpha_y, \omega_y, \alpha_z, \omega_z, \) and \( \Phi_z \) can be chosen to minimize the squared error between \( \hat{\mathbf{G}} \) and \( \mathbf{G} \). Unfortunately, analytical expressions are not available. Therefore, we determine some of the parameters by imposing meaningful properties of \( \mathbf{G}(t) \) to \( \hat{\mathbf{G}}(t) \):

1. first derivatives of \( G_x(t) \) and \( G_z(t) \) both zero at \( t = 0 \)
2. position and value of extrema of \( G_y(t) \)
which translates into

\[
\tan \Phi_a = -\frac{\alpha_a}{\omega_a}, \quad a \in \{x, z\}
\]  
\[
\omega_y = 2 \arctan (r_y)
\]  
\[
K_y = \left( \frac{4}{5} \right) \frac{1}{2r_y} \sqrt{1 + r_y^2} e^{\frac{1}{r_y} \arctan r_y},
\]  
with \( r_y = \frac{\omega_y}{\alpha_y} \) (analogous definitions for \( r_x \) and \( r_z \)). The remaining parameters \( \alpha_x, \omega_x, r_y, \alpha_z, \) and \( \omega_z \) are found by a grid-search minimization of the squared error between \( \hat{G} \) and \( G \). The parameters we obtained are \( r_x = 0.83, \omega_x = 1.884, r_y = 0.64, \omega_y = 1.139, r_z = 0.12, \) and \( \omega_z = 0.239 \). In Fig. 8.4, we plot \( G(t) \) and its estimate \( \hat{G}(t) \).

Using the following quantities

\[
A_{p,a,\tau} = e^{-\frac{\alpha_a}{\tau}} R(-\omega_a/\tau), \quad a \in \{x, y, z\}
\]  
\[
A_{f,a,\tau} = e^{-\frac{\alpha_a}{\tau}} R(-\omega_a/\tau), \quad a \in \{x, z\}
\]  
\[
A_{y,\tau} = e^{-\frac{\alpha_y}{\tau}} R(\omega_y/\tau)
\]  
\[
A_{p,\tau} = \text{diag}(A_{x,\tau}, A_{y,\tau}, A_{z,\tau}) \in \mathbb{R}^{6 \times 6}
\]  
\[
A_{f,\tau} = \text{diag}(A_{x,\tau}, A_{y,\tau}, A_{z,\tau}) \in \mathbb{R}^{6 \times 6}
\]  
\[
s_D = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T
\]  
\[
\tilde{C}_D = \begin{bmatrix} 
1 & \tan \Phi_x & 0 & 0 & 0 & 0 \\
0 & 0 & K_y & 0 & 0 \\
0 & 0 & 0 & 1 & \tan \Phi_z 
\end{bmatrix}
\]
we can generate \( \hat{B}^{(\tau)}_k \) with a 6th-order LSSM

\[
\hat{B}^{(\tau)}_k = \begin{cases} 
\lambda \text{RFC}_D (A_{p,\tau})^{|k|} s_D, & \text{for } k \leq 0 \\
\lambda \text{RFC}_D (A_{f,\tau})^k s_D, & \text{for } k > 0 
\end{cases}
\]
where \( C_D = H\tilde{C}_D \). As the state-transition matrices \( A_{p,\tau} \) and \( A_{f,\tau} \) depend on \( \tau \), we need one model per time scale.

**LSSM for External Magnetic Fields**

In addition to the magnetic field produced by the moving dipole, the magnetometer also measures additive magnetic fields from external sources among which is the Earth magnetic field. Assuming that the sensor is not moving while performing a movement (it may be moved in
between), those external magnetic fields result in a constant field vector denoted \( \mathbf{B}_1 \in \mathbb{R}^3 \). We directly include the constant component \( \mathbf{B}_1 \) (to be estimated), trivially modeled with a first order LSSM.

**Joint LSSM for Magnetic Field Measurements**

For a given time scale \( \tau \), the parameters of the joint 7th-order LSSM of the raw measurements are

\[
A_{p,\tau} = \text{diag}(A_{p,\tau}, 1) \in \mathbb{R}^{7 \times 7} \quad (8.39)
\]

\[
A_{f,\tau} = \text{diag}(A_{f,\tau}, 1) \in \mathbb{R}^{7 \times 7} \quad (8.40)
\]

\[
s = s_p = s_f = [s_D^T \ 1]^T \in \mathbb{R}^7 \quad (8.41)
\]

\[
C = C_p = C_f = [\lambda RFC_D \ \mathbf{B}_1]. \quad (8.42)
\]

Note that \( C \) can be written as

\[
C = \lambda RFC_D P_D + \mathbf{B}_1 P_I, \quad (8.43)
\]

with \( P_D = [I_6 \ 0_{6 \times 1}] \in \mathbb{R}^{6 \times 7} \) and \( P_I = [0_{1 \times 6} \ 1] \in \mathbb{R}^{1 \times 7} \).

### 8.2.3 Detecting and Estimating Movements of a Magnet

Given the magnetic field measurements \( y_1, \ldots, y_K \in \mathbb{R}^3 \), we want to detect and estimate straight-line movements of a magnet. Since our algorithm will run online and multiple gestures will be made, we use a local cost

\[
J_k(C, \tau) = \sum_{i=1}^{k} \gamma_{\tau}^{k-i} \| y_i - C A_{p,\tau}^{k-i} s \|^2 + \sum_{i=k+1}^{K} \gamma_{\tau}^{i-k} \| y_i - C A_{f,\tau}^{i-k} s \|^2, \quad (8.44)
\]

as in (6.1) with \( A_p = A_{p,\tau}, \ A_f = A_{f,\tau}, \ C_f = C_p = C \) (cf. (8.42)), \( s_f = s_p = s \), and \( \gamma_f = \gamma_p = \gamma_{\tau} \), for some \( \gamma_{\tau} \in (0, 1) \).

**Recursive Cost Computation**

The cost \( J_k(C, \tau) \) is efficiently computed using forward/backward recursions as described in Sec. 6.1.2 and writes as in (6.22), i.e.,

\[
J_k(C, \tau) = \kappa_k^{(\tau)} - 2 \text{tr} \left( C S_k^{(\tau)} \right) + \text{tr} \left( CW_k^{(\tau)} C^T \right), \quad (8.45)
\]


\[ \kappa_k(\tau) = \overrightarrow{\kappa}_k(\tau) + \overleftarrow{\kappa}_k(\tau) \]  
(8.46)

\[ \zeta_k(\tau) = \overrightarrow{\zeta}_k(\tau) + \overleftarrow{\zeta}_k(\tau) \]  
(8.47)

\[ W_k(\tau) = \overrightarrow{W}_k(\tau) + \overleftarrow{W}_k(\tau). \]  
(8.48)

Minimizing the Cost Function

For a fixed time scale \( \tau \) (we here drop the superscript \( \tau \)), we focus on minimizing

\[ J_k(C, \tau) = J_k(\lambda, R, d_m, B_I, \tau) \]  
(8.49)

with respect to \( C \) as in (8.42) with \( \lambda \in \mathbb{R}_+, R \in SO_3(\mathbb{R}), d_m \in S^3 \), and \( B_I \in \mathbb{R}^3 \).

The minimization over \( B_I \) is a “Constrained Offset” problem of Table 6.2. Thus, we have

\[ \hat{B}_I = (\xi_k^T - \lambda RFC_D P_D W_k)P_I^T (P_I W_k P_I^T)^{-1} \]  
(8.50)

and after simplifications, using “Post-Multiplied” of Table 6.2,

\[ J_k(\lambda, R, d_m) = \min_{B_I \in \mathbb{R}^3} J_k(\lambda, R, d_m, B_I, \tau) \]  
(8.51)

\[ = \tilde{\kappa}_k - 2\lambda \text{tr}(RF\tilde{\zeta}_k) + \lambda^2 \text{tr}(FW_k F_k^T), \]  
(8.52)

with

\[ \tilde{\kappa}_k = \kappa_k - \text{tr}(\xi_k^T P_I^T (P_I W_k P_I^T)^{-1} P_I \xi_k) \]  
(8.53)

\[ \tilde{\zeta}_k = C_D P_D (\xi_k - W_k P_I^T (P_I W_k P_I^T)^{-1} P_I \xi_k) \]  
(8.54)

\[ \tilde{W}_k = C_D P_D (W_k - W_k P_I^T (P_I W_k P_I^T)^{-1} P_I W_k) P_I^T C_D^T. \]  
(8.55)

Unfortunately, a joint minimization of the cost (8.52) with respect to \( \lambda \), \( R \), and \( d_m \) has no known closed-form solution. However, for a fixed moment direction \( d_m \), the optimization is a “Scaled Wahba’s Problem”, as described in Sec. 6.1.3. In order to minimize the cost in (8.52), we propose either a grid-search minimization with respect to the direction \( d_m \) or, alternatively, a coordinate descent algorithm with respect to the parameters \( (\lambda d_m, R) \), as described in the following.

For given \( (\lambda, d_m) \), the remaining minimization of (8.52) over \( R \in SO_3(\mathbb{R}) \) is a “Wahba’s Problem” (cf. Sec. 6.1.3).
For a fixed $R \in \mathcal{SO}_3(\mathbb{R})$, we now focus on the remaining minimization over $(\lambda, d_m)$. We express $F$, defined in (8.24), as

$$F = \langle d_m, d_r \rangle D_1 + \langle d_m, d_v \rangle D_2 + \langle d_m, d_{rv} \rangle D_3,$$

(8.56)

with $D_1 = \text{diag}(1, 1, 0)$, $D_3 = \text{diag}(0, 0, 1)$, and

$$D_2 = \begin{bmatrix}
0 & 1 & 0 \\
-1 & 0 & -1 \\
0 & 0 & 0
\end{bmatrix}.\)

(8.57)

Plugging this expression in the cost function of (8.52) leads to

$$\tilde{J}_k(R, \tilde{m}) = \tilde{\kappa}_k - 2\tilde{m}^T b + \tilde{m}^T W_D \tilde{m},$$

(8.58)

with, for all $(i, j) \in \{1, 2, 3\}^2$,

$$\{W_D\}_{i,j} = \text{tr}(D_i \tilde{W}_k D_j^T)$$

(8.59)

$$\{b\}_i = \text{tr}(RD_i \tilde{\xi}_k)$$

(8.60)

$$\tilde{m} = \lambda \left[ \langle d_m, d_r \rangle \quad \langle d_m, d_v \rangle \quad \langle d_m, d_{rv} \rangle \right]^T.$$  

(8.61)

The matrix $W_D \in \mathbb{R}^{3 \times 3}$ is symmetric positive semi-definite since for any $\tilde{m} \in \mathbb{R}^3$, $\tilde{m}^T W_D \tilde{m} = \lambda^2 \text{tr}(FWF^T) \geq 0$. Thus, we have

$$\arg\min_{\tilde{m} \in \mathbb{R}^3} \tilde{J}_k(R, \tilde{m}) = W_D^{-1} b$$

(8.62)

$$\min_{\tilde{m} \in \mathbb{R}^3} \tilde{J}_k(R, \tilde{m}) = \tilde{\kappa}_k - b^T W_D^{-1} b,$$

(8.63)

which concludes the coordinate descent algorithm.

**Detection and Estimation**

In order to detect a gesture at time $k$, for a given time-scale $\tau$, given the offset magnetic field vector $\hat{B}_1$ estimated with the joint LSSM, we use the hypothesis test

- $H_0: \lambda = 0$
- $H_1: \lambda > 0$, $R \in \mathcal{SO}_3(\mathbb{R})$, $d_m \in S^3$

and the following log-likelihood ratio (cf. Sec. 7.2.1)

$$\text{LLR}^{(\tau)}_k = -\frac{1}{2} \ln \left( \frac{\min_{\lambda, R, d_m} J_k(\lambda, R, d_m, \hat{B}_1, \tau)}{J_k(\lambda = 0, R, d_m, \hat{B}_1, \tau)} \right),$$

(8.64)
where the minimization in the numerator is under $\mathcal{H}_1$. The denominator does not depend on $R$ and $d_m$ due to $\lambda = 0$.

In order to take several time scales into account, we discretize the parameter $\tau \in \{\tau_j, j \in J \subset \mathbb{N}\}$. Each $\tau$ has its own LSSM and the final quantity we use is

$$LLR_k = \max_{j \in J} LLR^{(\tau_j)}_k.$$  \hspace{2cm} (8.65)

If $LLR_k$ is above a given threshold and locally maximum then a gesture is detected and the estimated gesture parameters can be retrieved using the method described in Sec. 8.2.3.

### 8.2.4 Results

First, we test our algorithm on simulated data generated using the dipole equation (8.18) with additive white Gaussian noise of variance $6.25 \cdot 10^{-2} \mu\text{T}^2$, a magnet with $\mu_{\text{am}} = 4 \cdot 10^{-4} \text{J}/\mu\text{T}$, and a magnetometer of sampling frequency $f_s = 40 \text{ Hz}$. We use three cases

1. close and slow: $r = 3 \text{ cm}$, $v = 30 \text{ cm/s}$,
2. close and fast: $r = 3 \text{ cm}$, $v = 60 \text{ cm/s}$,
3. far and slow: $r = 6 \text{ cm}$, $v = 30 \text{ cm/s}$.

For each scenario, we vary the directions $d_r$, $d_v$, and $d_m$ to produce 10000 gestures. We here assume that $(d_m, d_v) \approx 0$ and thus we have

$$d_m \approx \cos \phi \ d_r + \sin \phi \ d_{rv},$$  \hspace{2cm} (8.66)

for some $\phi \in (0, 2\pi)$. For the algorithm, $\tau$ and $\phi$ are uniformly discretized respectively from 1 to 30 with a step of 1 and from 0 to 90° with a step of 0.5°. The results are summarized in Table 8.1 which shows very good detection, localization, and estimation abilities. $(d, \hat{d})$ denotes the angle between the direction $d$ and the estimated direction $\hat{d}$. $|\ell - \hat{\ell}|$ is the time index shift between the time and estimated time of a gesture detection.

The algorithm is also tested on an iPhone 5 with iOS7, which embeds a AK8975 magnetometer of sampling frequency $f_s = 40 \text{ Hz}$. We use an Apple earphone (with $\mu_{\text{am}} = 4 \cdot 10^{-4} \text{J}/\mu\text{T}$) as magnet. Fig. 8.5 shows a typical observed signal and its estimate. Even with the use of a weak magnet, the algorithm demonstrates promising results for tracking human-made gestures.
### Table 8.1 – Average errors for 10000 simulated gestures

| Errors | $|\ell - \hat{\ell}|$ | $|\frac{r - \hat{r}}{r}|$ (%) | $(d_r, \hat{d}_r)$ | $|\frac{v - \hat{v}}{v}|$ (%) | $(d_v, \hat{d}_v)$ | $|\phi - \hat{\phi}|$ |
|--------|-----------------|-----------------|----------------|-----------------|----------------|----------------|
| Case 1 | 0               | 1.0             | 2.2°           | 1.9             | 0.8°           | 1.1°           |
| Case 2 | 0               | 0.8             | 2.4°           | 1.2             | 1.2°           | 1.3°           |
| Case 3 | 0.04            | 3.8             | 11.2°          | 11.8            | 5.2°           | 9.1°           |

#### Figure 8.5 – Raw magnetic field measurements and estimated signals.

### 8.3 Detecting and Estimating Electric Dipole Movements From Electrode Measurements

Several physical phenomena can be well modeled as a uniform straight-line movement of an electric dipole. For instance, the depolarization of cells that is a common phenomenon observed in biology for transmitting stimuli, can be seen as such. A cell depolarization creates transient electrical potential differences measurable by nearby electrodes, as observed in the ECG resulting from depolarization of heart muscle cells. Given voltage measurements, we want to infer the speed and direction of the depolarization process. This part extends [86].
8.3.1 Physical Model of Voltage Measurements

An electric dipole of moment \( p \in \mathbb{R}^3 \) creates at an electrode point placed at \( r \in \mathbb{R}^3 \) from the dipole, a potential

\[
\phi(r) = \frac{1}{4\pi\epsilon} \frac{\langle p, r \rangle}{\|r\|^3}, \tag{8.67}
\]

where \( \epsilon \) is the permittivity of the medium (assumed homogeneous). Quantities in \( \mathbb{R}^3 \) are highlighted in bold. The gradient of the potential

\[
\nabla \phi(r) = -\frac{1}{4\pi\epsilon} \left( \frac{3\langle p, r \rangle r}{\|r\|^5} - \frac{p}{\|r\|^3} \right) = -E(r) \tag{8.68}
\]

is nothing else than the negative electric field created by the dipole at the electrode point. It follows that for a small displacement \( \delta r \) compared to \( r \), the first-order Taylor approximation of the potential is

\[
\phi(r + \delta r) \approx \phi(r) - \langle E(r), \delta r \rangle. \tag{8.69}
\]

Now, assume that an electric dipole is moving uniformly on a straight line while keeping the electric dipole moment constant. Using a coordinate system of a sensor with origin \( O \), the position \( P_k \) of the dipole at time index \( k \in \mathbb{Z} \) is

\[
\overrightarrow{OP_k} = \overrightarrow{r_k} = r \overrightarrow{d_r} + kv \overrightarrow{d_v} = r (\overrightarrow{d_r} + (k/\tau) \overrightarrow{d_v}), \tag{8.70}
\]

where \( r \in \mathbb{R}_+^* \) is the minimum distance to \( O \), \( \overrightarrow{d_r} \in S^3 \) the direction at this minimum distance, \( v \in \mathbb{R}_+^* \) the per-sample speed, \( \overrightarrow{d_v} \in S^3 \) the direction of movement, and \( \tau = r/v \).

Let \( A \) denote an electrode point, and \( \overrightarrow{d_A} = \overrightarrow{OA}/\|\overrightarrow{OA}\| \). The potential \( \phi_k(A) \) induced by the electric dipole at \( A \) and time index \( k \in \mathbb{Z} \) is

\[
\phi(\overrightarrow{P_kA}) = \phi(\overrightarrow{OA} - \overrightarrow{r_k}). \tag{8.71}
\]

Assuming that \( \|\overrightarrow{OA}\| \ll r \), it follows that \( \|\overrightarrow{OA}\| \ll \|r_k\| \), for all \( k \in \mathbb{Z} \). Then, the first-order Taylor expansion of the potential measured at \( A \) is

\[
\phi_k(A) = \phi(-\overrightarrow{r_k}) - \langle E(\overrightarrow{r_k}), \overrightarrow{OA} \rangle. \tag{8.72}
\]

In the following, we assume that the multi-channel electrode used for the measurements is designed such that all electrode points are close
Example Applications

enough to a reference point $O$, i.e., $\|\overrightarrow{OA}\| \ll r$ and that each voltage measurement between two points $A-B$ satisfies $\|\overrightarrow{AB}\| \ll r$. It results that the voltage measured by the electrode $A-B$ is (first-order Taylor expansion)

$$\phi_k(A) - \phi_k(B) = \langle \mathbf{E}(r_k), \overrightarrow{AB} \rangle. \quad (8.73)$$

Thus, the measured voltage is the projection of a 3D discrete-time signal into the electrode direction $\overrightarrow{AB}$. Furthermore, the electric field $\mathbf{E}(r_k)$ generated by an electric dipole has the same expression as the magnetic field $\mathbf{B}(r_k)$ induced by a magnetic dipole as in (8.18) (replace $m = m_d$ by $p = p_d$). It follows that

$$\phi_k(A) - \phi_k(B) = \lambda \langle \overrightarrow{AB}, RFHG(k/\tau) \rangle, \quad (8.74)$$

with $\lambda = \frac{p}{4\pi\epsilon r^3} \in \mathbb{R}^+$, $G(t)$, $H$, $F$, $R$ as in, respectively, (8.22), (8.23), (8.24), and (8.25). Unlike the magnetometer in Sec. 8.2 that projects the magnetic field in a 3D-orthonormal basis, a voltage measurement is only one projection of the electric field. In order to recover the dipole movement in 3D from measurements, a multi-channel electrode is required with at least 3 linearly independent electrode directions. In [86], we chose a tetrahedral electrode [79]. We here assume that we have $M \in \mathbb{N}$ voltage measurements and we define the matrix $S \in \mathbb{R}^{3 \times M}$ containing in its columns the electrode directions. Combining the $M$-channel measurements, we obtain

$$V_k = \lambda S^T RFHG(k/\tau). \quad (8.75)$$

**8.3.2 LSSM for Measurement Signals**

**LSSM for the Electric Field of the Dipole**

Using the same notation and model for $G(t)$ as in Sec. 8.2, we can generate $\hat{V}_k(\tau)$ with a 6th-order LSSM

$$\hat{V}_k(\tau) = \begin{cases} 
\lambda S^T RFC_D(A_{p,\tau}^D)^{|k|} s_D, & \text{for } k \leq 0 \\
\lambda S^T RFC_D(A_{f,\tau}^D)^k s_D, & \text{for } k > 0.
\end{cases} \quad (8.76)$$

As the state-transition matrices $A_{p,\tau}^D$ and $A_{f,\tau}^D$ depend on $\tau$, we need one model per time scale.
8.3 Detecting and Estimating Electric Dipole Movements From Electrode Measurements

LSSM for Interference Signals

In practical applications, electrodes often experience disturbances from external sources (e.g., in medical use, breathing or body movements), which create interference signals. Similarly as in Sec. 8.1.1, we locally model additive interferences with discrete-time polynomials of degree $n_I-1$ with LSSM parameters $A_I \in \mathbb{R}^{n_I \times n_I}$, $s_I \in \mathbb{R}^{n_I}$ (as in Sec. 8.1.1), and an observation vector depending on the coefficients of the polynomial. In agreement with the LSSM of the dipole signals, we use one two-sided polynomial for each channel measurement and denote $C^p_I$, $C^f_I \in \mathbb{R}^{M \times n_I}$ their observation matrices. Besides, at the meeting point of each two-sided polynomial we impose smoothness constraints such as continuity and differentiability of the counterpart continuous signals. Such linear constraints can be expressed as

$$[C^p_I \quad C^f_I] = C_I B,$$  \hspace{1cm} (8.77)

with $B \in \mathbb{R}^{d \times 2n_I}$ a fixed matrix whose row vectors span the subspace of constraints and $C_I \in \mathbb{R}^{M \times d}$ containing the coefficients to be estimated. For instance, if we want to enforce continuity of the interference signals, we simply have $d = n_I$ and

$$B = [I_{n_I} \quad I_{n_I}].$$  \hspace{1cm} (8.78)

Joint LSSM for the Measurements

For a given time scale $\tau$, the parameters of the joint two-sided $(6 + n_I)$th-order LSSM of the raw measurements are

$$C_p = [\lambda S^T RFC_D \quad C^p_I] \in \mathbb{R}^{M \times (6+n_I)}$$  \hspace{1cm} (8.79)

$$A_{p,\tau} = \text{diag}(A^D_{p,\tau}, A^{-1}_I) \in \mathbb{R}^{(6+n_I) \times (6+n_I)}$$  \hspace{1cm} (8.80)

$$C_f = [\lambda S^T RFC_D \quad C^f_I] \in \mathbb{R}^{M \times (6+n_I)}$$  \hspace{1cm} (8.81)

$$A_{f,\tau} = \text{diag}(A^D_{f,\tau}, A_I) \in \mathbb{R}^{(6+n_I) \times (6+n_I)}$$  \hspace{1cm} (8.82)

$$s = s_p = s_f = [s^T_D \quad s^T_I]^T \in \mathbb{R}^{(6+n_I)}.$$  \hspace{1cm} (8.83)

Note that $C = [C_p \quad C_f]$ can be decomposed into

$$C = \lambda S^T RFC_D P_D + C_I B P_I,$$  \hspace{1cm} (8.84)

with

$$P_D = \begin{bmatrix} I_6 & 0_{6 \times n_I} & I_6 & 0_{6 \times n_I} \end{bmatrix}$$  \hspace{1cm} (8.85)

$$P_I = \begin{bmatrix} 0_{n_I \times 6} & I_{n_I} & 0_{n_I \times 6} & 0_{n_I \times n_I} \\ 0_{n_I \times 6} & 0_{n_I \times n_I} & 0_{n_I \times 6} & I_{n_I} \end{bmatrix}.$$  \hspace{1cm} (8.86)
8.3.3 Detecting and Estimating Electric Dipole Movements

Given voltage measurements \( y_1, \ldots, y_K \in \mathbb{R}^M \), at any time \( k \), we want to test whether a dipole movement occurred and estimate its parameters. Since our algorithm incorporates a polynomial model, we use a local cost

\[
J_k(C, \tau) = \sum_{i=1}^{k} \gamma_{\tau}^{k-i} \| y_i - C_p A_p^{k-i} s \|^2 + \sum_{i=k+1}^{K} \gamma_{\tau}^{i-k} \| y_i - C_f A_f^{i-k} s \|^2, \tag{8.87}
\]

as in (6.1) with \( A_p = A_{p,\tau}, A_f = A_{f,\tau}, C_f \) as in (8.81), \( C_p \) as in (8.79), \( s_f = s_p = s \), and \( \gamma_f = \gamma_p = \gamma_{\tau} \), for some \( \gamma_{\tau} \in (0, 1) \).

Recursive Cost Computation

The cost \( J_k(C, \tau) \) is efficiently computed using forward/backward recursions as in Sec. 6.1.2 and parametrized as in (6.22) with \( \kappa_k(\tau), \xi_k(\tau) \), and \( W_k^{(\tau)} \) as in (6.23), (6.24), and (6.25), respectively.

Minimizing the Cost Function

For a fixed time scale \( \tau \) (we here drop the superscript \( \tau \)), we focus on minimizing \( J_k \) with respect to \( C \) as in (8.84) with \( \lambda \in \mathbb{R}_+, R \in SO_3(\mathbb{R}), d_p \in S^3 \), and \( C_I \in \mathbb{R}^{M \times d} \).

The minimization over \( C_I \in \mathbb{R}^{M \times d} \) is a “Constrained Offset” problem of Table 6.2. Thus, we have

\[
\hat{C}_I = (\xi_k^T - \lambda S^T R F C_D P_D W_k) P_I^T B^T (B P_I W_k P_I^T B^T)^{-1}, \tag{8.88}
\]

and after simplifications, using “Post-Multiplied” of Table 6.2

\[
\tilde{J}_k(\lambda, R, d_p) = \min_{C_I \in \mathbb{R}^{M \times d}} J_k(\lambda, R, d_p, C_I, \tau) \tag{8.89}
\]

\[
= \bar{\kappa}_k - 2\lambda \text{tr}(RF\xi_k) + \lambda^2 \text{tr}(S^T RF \hat{W}_k F^T R^T S), \tag{8.90}
\]

with \( \hat{P}_I = BP_I \) and

\[
\bar{\kappa}_k = \kappa_k - \text{tr}(\xi_k^T \hat{P}_I^T (\hat{P}_I W_k \hat{P}_I)^{-1} \hat{P}_I \xi_k) \tag{8.91}
\]

\[
\hat{\xi}_k = C_D P_D (\xi_k - W_k \hat{P}_I^T (\hat{P}_I W_k \hat{P}_I)^{-1} \hat{P}_I \xi_k) S^T \tag{8.92}
\]

\[
\hat{W}_k = C_D P_D (W_k - W_k \hat{P}_I^T (\hat{P}_I W_k \hat{P}_I)^{-1} \hat{P}_I W_k) P_D C_D^T. \tag{8.93}
\]
Unfortunately, a joint minimization of the cost in (8.90) with respect to \( \lambda \in \mathbb{R}_+ \), \( R \in SO_3(\mathbb{R}) \), and \( d_p \in S^3 \) has no known closed-form solution. Thus, we propose a coordinate descent algorithm with respect to \((\lambda, d_p, R)\).

For given \((\lambda, d_p)\), minimizing (8.90) over \( R \in SO_3(\mathbb{R}) \) is a “Weighted Orthogonal Procrustes Problem” (cf. Sec. 6.1.3). In case \( SS^T \approx \eta I \), for some \( \eta \in \mathbb{R} \), is a good approximation, it simplifies into an “Orthogonal Procrustes Problem”.

For a fixed \( R \in SO_3(\mathbb{R}) \), we now focus on minimizing (8.90) over \((\lambda, d_p)\). Similarly to Sec. 8.2.3 (8.90) writes as
\[
\tilde{J}_k(R, \tilde{p}) = \kappa_k - 2\tilde{p}^Tb + \tilde{p}^TW_D\tilde{p},
\]
with, for all \((i, j) \in \{1, 2, 3\}^2\),
\[
\{W_D\}_{i,j} = \text{tr}\left(S^T R D_i \tilde{W}_k D_j^T R^T S\right) \quad (8.95)
\]
\[
\{b\}_i = \text{tr}\left(R D_i \tilde{\xi}_k\right) \quad (8.96)
\]
\[
\tilde{p} = \lambda \left[ \langle d_p, d_r \rangle \langle d_p, d_v \rangle \langle d_p, d_{rv} \rangle \right]^T. \quad (8.97)
\]
The optimum \( \tilde{p} \) is found as in (8.62).

**Detection and Estimation**

In order to detect a moving dipole at time \( k \), for a given time-scale \( \tau \), given the polynomial coefficients \( \hat{C}_1 \) estimated with the joint LSSM, we use the hypothesis test

- \( \mathcal{H}_0: \lambda = 0 \)
- \( \mathcal{H}_1: \lambda > 0, R \in SO_3(\mathbb{R}), d_p \in S^3 \)

and the following log-likelihood ratio (cf. Sec. 7.2.1)
\[
\text{LLR}^{(\tau)}_k = -\frac{1}{2} \ln \left( \frac{\min_{\lambda, R, d_p} J_k(\lambda, R, d_p, \hat{C}_1, \tau)}{J_k(\lambda = 0, R, d_p, \hat{C}_1, \tau)} \right), \quad (8.98)
\]
where the minimization in the numerator is under \( \mathcal{H}_1 \). The denominator does not depend on \( R \) and \( d_p \) due to \( \lambda = 0 \).

In order to take several time scales into account, we discretize the parameter \( \tau \in \{\tau_j, j \in \mathcal{J} \subset \mathbb{N}\} \). Each \( \tau \) has its own LSSM and the final quantity we use is
\[
\text{LLR}_k = \max_{j \in \mathcal{J}} \text{LLR}^{(\tau_j)}_k. \quad (8.99)
\]
If $\text{LLR}_k$ is above a given threshold and locally maximum then a moving dipole is detected and the estimated movement parameters are retrieved using the method described in Sec. 8.3.3.

### 8.3.4 Results

We here present the same results as in [86]. We assume that the depolarization process is such that the electric dipole is aligned with the direction of movement, i.e., $\mathbf{d}_v = \pm \mathbf{d}_p$. Moreover, we use a tetrahedral electrode $ABCD$ such that $ABC$ is an equilateral triangle and $D$ a summit with $AD = BD = CD$. In this case, the assumption $S^T S \approx \eta I$ is a good approximation. Fig. 8.6 is a schematic representation of the tetrahedral electrode and an electrode dipole movement.

**Performance on Simulated Data**

We test our algorithm on simulated data generated using the dipole equation (8.67) with additive white Gaussian noise of variance $\sigma^2 = 10^{-4}$ V$^2$, a dipole with $\frac{\mu}{4\pi \varepsilon} = 70$ V · m$^2$, and a tetrahedral electrode of sampling frequency of 960 Hz. We use three different cases.
8.3 Detecting and Estimating Electric Dipole Movements From Electrode Measurements

1. \( r = 3 \text{ cm}, \ v = 4.1 \text{ m/s}, \)

2. \( r = 3 \text{ cm}, \ v = 5.8 \text{ m/s}, \)

3. \( r = 6 \text{ cm}, \ v = 5.8 \text{ m/s}. \)

For each case we vary the directions \( \mathbf{d}_r \) and \( \mathbf{d}_v \) to produce 7000 dipole movements. For the algorithm, \( \tau \) is uniformly sampled from 3 to 16 with a step size of 1 and a polynomial of degree 2 is used to model interferences. The simulations are performed both without and with interference signals consisting of additive constants and cosines of different frequencies on each channel (respectively 4 Hz, 6 Hz, and 8 Hz).

As shown in Table 8.2, the algorithm demonstrates a good estimation ability even in the presence of these interference signals. Note that \( (\mathbf{d}, \hat{\mathbf{d}}) \) denotes the angle between \( \mathbf{d} \) and \( \hat{\mathbf{d}} \).

**Table 8.2 — Average errors for 7000 simulated dipole movements with (blue) and without (black) interferences.**

| Errors   | \( \frac{|r - \hat{r}|}{r} \) (%) | \( (\mathbf{d}_r, \hat{\mathbf{d}}_r) \) | \( \frac{|v - \hat{v}|}{v} \) (%) | \( (\mathbf{d}_v, \hat{\mathbf{d}}_v) \) |
|----------|----------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Case 1   | 6.0                             | 6.2°                            | 7.7°                            | 4.8°                            |
| Case 2   | 5.1                             | 5.8°                            | 7.4°                            | 5.3°                            |
| Case 3   | 2.8                             | 1.5°                            | 16.4°                           | 5.5°                            |

**Performance on Experimental Data**

We also test our algorithm on experimental data. A sensor prototype including a tetrahedral electrode \[79\] is placed inside a pot filled with saline water. The dipole consisting of two close-by electrodes charged with 2 V is attached to a string and pulled by a motor. Fig. 8.7 shows a typical signal recorded while the dipole is moving and the estimated signals reconstructed with the LSSM from our algorithm. It is especially worth noticing the splitting of the signal in dipole model and polynomial part.

In Fig. 8.8, we investigate the influence of an increase of the distance \( r \). The black point represents the circumcenter of the electrodes and the green dashed lines the circle of radius with correct distance. The change of distance is accurately captured by our algorithm. The variations of \( \mathbf{d}_r \) are mainly caused by the experimental setup and not by the algorithm.
8.4 Detection and Estimation of a Modulated Signal

We want to detect pulses of sinusoidal shape of frequency $\Omega \in \mathbb{R}_+$ in an amplitude-modulated carrier signal of frequency $\Omega_g \in \mathbb{R}_+$ and affected by additive white Gaussian noise. A typical observed signal is displayed in Fig. 8.9 upper plot.
8.4 Detection and Estimation of a Modulated Signal

The carrier signal is seen as an interference signal \( g(C_g) \) with LSSM parameters

\[
A_{f,g} = A_{p,g}^{-1} = R(\Omega_g) \tag{8.100}
\]

\[
s_{f,g} = s_{p,g} = [1 \quad 0]^T, \tag{8.101}
\]

and unknown

\[
C_g = [C_{p,g} \quad C_{f,g}] \in \mathbb{R}^{1 \times 4}. \tag{8.102}
\]

The signal of interest \( f(C) \), consisting of pulses of sinusoidal shape, is generated with the LSSM parameters

\[
A_f = A_p^{-1} = R(\Omega) \tag{8.103}
\]

\[
s_f = s_p = [1 \quad 0]^T, \tag{8.104}
\]

and unknown

\[
C = [C_p \quad C_f] \in \mathbb{R}^{1 \times 4}. \tag{8.105}
\]

Figure 8.9 – Synthetic example of amplitude-modulated pulse detection.
For all time indices $k$, we recursively compute a cost $J_k(C, C_g)$ as in \((7.20)\) with a two-sided exponential window with parameters $s_w = C_w = 1$ and $A_{p,w} = A_{f,w} = \gamma$ for some $\gamma \in (0, 1)$.

In Fig. 8.9, we illustrate the results of our signal estimation method. In the lower plot, we display the log-likelihood ratio (cf. \((7.24)\))

$$LLR_k = -\frac{1}{2} \ln \left( \frac{\min_{C,C_g} J_k(C, C_g)}{J_k(C = 0, C_g = 0)} \right), \quad (8.106)$$

which indicates how likely the presence of a signal of interest is. In the middle part of Fig. 8.9, we plot the estimated signal obtained at index $k = 350$ where $LLR_k$ is maximum. Note the actual separation of the carrier signal from the signal of interest.

### 8.5 Detection and Classification of Fires

Most of the current fire detectors raise a fire alarm as soon as the recorded signals exhibit characteristic features such as high amplitudes, signal ratios in a certain range, or vigorous fluctuations. Such signal features typically occur when a fire actually arises but also when other nuisance phenomena happen, such as the emergence of steam while cooking. However, a more careful analysis of the signals recorded from a fire detector might succeed in robustly distinguishing fires from nuisances and classifying fires according to common characteristics. Based on a multi-channel signal as in Fig. 8.10 recorded with an optical smoke detector, we design an algorithm which robustly detects and classifies fires and distinguishes them from nuisances.

This example application is described in detail in [63] but is presented here for illustrative purposes.

#### 8.5.1 Physical Model of the Optical Smoke Detector

An optical smoke detector uses the physical phenomenon of light scattering. One or several light sources emit an optical beam with a given wavelength (often infrared or blue). Several photosensors measure the received light intensity (in a given wavelength range) that is scattered towards them. This intensity depends on the location of the sensor with respect to the light source but also on the type and number of smoke particles that scatter the light beam.
8.5 Detection and Classification of Fires

When a fire or a nuisance occurs, a specific type of smoke particle is progressively created and some of the smoke particles gradually enter the fire detector chamber. Upon such an event, the complex physical model can be simplified with the following assumptions.

1. Inside the fire detector chamber and for a short period of time, the number of smoke particles grows linearly with time.

2. The received intensity depends linearly on the number of particles. Furthermore, this linear dependency is characterized by the type of particles, the wavelength of the light source, and the location of the photosensor.

This physical phenomena along with its simplified model essentially summarizes as follows. When a fire or a nuisance occurs, the recorded signals grow linearly. Moreover, for a given type of particle (i.e., a given class of fire or nuisance), the growth coefficients are constant, up to a scale factor corresponding to the number of particles.
8.5.2 LSSM for the Measured Signals

In the light of the simplified physical model, we use two-sided LSSM signals consisting of one line per channel, i.e., \( n_f = n_p = 2 \),

\[
A_f = A_p^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} 
\]

\( (8.107) \)

\[
s_f = s_p = [0 \ 1]^T, \quad (8.108) \]

\( C_p \in \mathbb{R}^{M \times 2} \), and \( C_f \in \mathbb{R}^{M \times 2} \). While \( \{C_p\}_1 \) and \( \{C_p\}_2 \) (first and second column of \( C_p \)) correspond to, respectively, the slopes and offsets of the left-sided lines, \( \{C_f\}_1 \) and \( \{C_f\}_2 \) (first and second column of \( C_f \)) correspond to, respectively, the slopes and offsets of the right-sided lines.

8.5.3 Detection and Classification of Fires and Nuisances

We here use the approach described in Sec. 7.3 with the use of a classifier. Given the measurements \( y_1, \ldots, y_K \in \mathbb{R}^M \) from the optical smoke detector, we want to detect and classify occurrences of fires and nuisances. We use the local cost \( J_k(C) \) as in \( (6.1) \) with an exponential window with \( (\gamma_p, \gamma_f) \in (0, 1)^2 \).

**Detection of Fires and Nuisances**

The detection of an event is based on a hypothesis test with the following classes. The class \( C_1 \) accounts for a fire or nuisance event. Thus, the target signal is constant before an event and has a linear increase after that event (typically a different growth per channel), which translates into

\[
C_1 = \left\{ C \in \mathbb{R}^{M \times 4} : \{C_p\}_2 = \{C_f\}_2, \{C_p\}_1 = 0, \{C_f\}_1 \in \mathbb{R}^M \right\} \quad (8.109)
\]

\[
= \{ \tilde{C}P_1 \in \mathbb{R}^{M \times 4} : \tilde{C} \in \mathbb{R}^{M \times 2} \}, \quad (8.110)
\]

with

\[
P_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (8.111)
\]

On the other hand, the class \( C_0 \) accounts for no event. Consequently, the target signal is constant (one constant per channel), which translates
into
\[ C_0 = \left\{ C \in \mathbb{R}^{M \times 4} : \{C_p\}_2 = \{C_f\}_2 , \{C_p\}_1 = \{C_f\}_1 = 0 \right\} \] (8.112)
\[ = \{ \tilde{C} P_0 \in \mathbb{R}^{M \times 4} : \tilde{C} \in \mathbb{R}^{M \times 1} \}, \] (8.113)
with
\[ P_0 = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}. \] (8.114)

Then, we use a log-likelihood ratio \( \text{LLR}_k \) as defined in (7.13), which peaks whenever there is an estimated change from no slopes to significant slope values. Note that the two minimization problems in the \( \text{LLR} \) expression are solved using the “Post-Multiplied” solution (cf. Sec. 6.1.3). If \( \text{LLR}_k \) is above a given threshold and locally maximum then an event (fire or nuisance) is detected.

### Classification of Fires and Nuisances

We consider 5 classes of fires, each of them emitting a characteristic type of smoke particle, and one class of nuisance which emits dust. Thus, the type \( T \in \mathcal{T} \) of each fire or nuisance belongs to
\[ \mathcal{T} = \{ \text{TF1, TF2, TF3, TF4, TF5, Dust} \}. \] (8.115)

If needed, more classes of fires or nuisances can be considered.

Assume that we have a labeled dataset of recordings, where one recording corresponds to the signals measured by the optical detector when a fire or a nuisance occurs. For each recording \( j \in \{1, \ldots, J\} \), we compute the likelihood ratio, as described above. At the time index where the likelihood ratio is maximum (i.e., when the fire or nuisance is assumed to start), we extract the estimated right-sided slope values
\[ a^{(j)}_f = \{ \hat{C}_f \}_1 \in \mathbb{R}^M, \] (8.116)
where \( \hat{C}_f \) is obtained by minimizing the cost function \( J_k(C) \) (on the \( j^{th} \) recording at the index of detection) with respect to \( C \in \mathcal{C}_1 \).

Referring to the simplified physical model, all the right-sided slopes \( a^{(j)}_f \in \mathbb{R}^M \) corresponding to a given type \( T \in \mathcal{T} \) should lie (approximately) on a line in the \( M \)-dimensional space. For each type \( T \in \mathcal{T} \), we learn the direction \( v_T \) of the line by extracting a one-dimensional subspace using the PCA method.
Figure 8.11 — Estimated right-sided slopes for 4-channel signals (only three coordinates are plotted), when a fire or nuisance arises. One point corresponds to the estimated right-sided slopes (at the time of an event) of one sample recording. Each line corresponds to a class $T \in \mathcal{T}$.

Thus, when confronted with a recorded signal, upon detecting a peak in the LLR and after having computed

$$
\hat{C} = \arg\min_{C \in C_1} J_k(C),
$$

we extract the estimated right-sided slopes (i.e., third column of $\hat{C}$) and assign the point (in $\mathbb{R}^M$) consisting of these slopes to the closest line among the ones generated by $v_T$, $T \in \mathcal{T}$. Finally, based on this closest line, we classify the detected fire or nuisance accordingly.

### 8.5.4 Results

In Fig. 8.10, we display typical signals measured by an optical smoke detector consisting of $M = 4$ channels when a specific fire occurs. The log-likelihood ratio accurately detects the beginning of a fire at a time of 48 s. Furthermore, at that instant, the estimated lines $\hat{y}^{(m)}$, $m \in \{1, \ldots, 4\}$,
correctly capture the dynamic of the signals $y^{(m)}$, $m \in \{1, \ldots, 4\}$. The estimated right-sided slope values are further used for classification.

In Fig. 8.11 for each type $T \in \mathcal{T}$, we display its estimated line of direction $v_T$ and its associated data points. Each data point corresponds to the estimated right-sided slopes of a sample recording. Note that we only display the first three coordinates of the four-dimensional points. Thus, the separation of the lines is even more apparent in $\mathbb{R}^4$. Using the classification described above, we correctly classify all the fires and the nuisance.

### 8.6 Detection and Estimation of Model Switches

We here provide two straightforward examples of model switch detection and estimation as described in Sec. 7.4.

#### 8.6.1 Piecewise Lines With Varying Noise Variance

Consider a discrete-time signal defined piecewise where each piece is a line with its own slope and offset and with its own noise variance. A typical observed signal is depicted in Fig. 8.12, upper plot, blue line.

The task is to locate the model switches and at each switch to estimate the right-sided and left-sided lines and noise variances. For that purpose, we use a two-sided LSSM signal with parameters $n_f = n_p = 2$,

$$A_f = A_p^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$  \hspace{1cm} (8.118)

$$s_f = s_p = [0 \ 1]^T,$$  \hspace{1cm} (8.119)

and unknown $C_p \in \mathbb{R}^{1 \times 2}$ and $C_f \in \mathbb{R}^{1 \times 2}$. We use the local cost $J_k(C)$ as in (6.1) with an exponential window with $\gamma_p = \gamma_f = 0.975$.

The detection of the model switches is based on the test

- $\mathcal{H}_0$: $C_p = C_f$, $\sigma_p^2 = \sigma_f^2$

- $\mathcal{H}_1$: $C_p \in \mathbb{R}^{1 \times 2}$, $\sigma_p^2 \in \mathbb{R}_+$, $C_f \in \mathbb{R}^{1 \times 2}$, $\sigma_f^2 \in \mathbb{R}_+$,

with a log-likelihood ratio as in (7.41) with

$$C_0 = \{ C = [C_p \ C_f] \in \mathbb{R}^{1 \times 4} : C_p = C_f \}$$  \hspace{1cm} (8.120)

$$C_p = C_f = \mathbb{R}^{1 \times 2}.$$  \hspace{1cm} (8.121)
Figure 8.12 — Example of a piecewise linear signal with piece-dependent additive noise. For each detected switch (vertical green dashed line), a few samples of the estimated right-sided line are plotted in purple while a few samples of the estimated left-sided line are plotted in orange.

The log-likelihood ratio is plotted in Fig. 8.12, lower plot, red line. The green dashed lines indicate the local maxima (within a certain window and above a given threshold) of \( LLR_k \), which indicates the presence of a model switch. Furthermore, for each model switch that we detect, we also plot few samples of the estimated right-sided line (in magenta) and left-sided line (in orange), on top of the observed signal.

Our algorithm is able to compensate for the change of noise variances. In this example, all model switches are correctly detected but note that the estimated locations may vary by a few time indices compared to the actual ones.

8.6.2 Piecewise Sinusoids With Varying Noise Variance

Consider a discrete-time signal defined piecewise where each piece is a sinusoid with its own amplitude and phase and with its own noise variance. A typical observed signal is depicted in Fig. 8.13, upper plot, blue line.
Figure 8.13 – Example of a piecewise sinusoidal signal with piece-dependent noise. For each detected switch (vertical green dashed line), a few samples of the estimated right-sided sinusoid are plotted in purple while a few samples of the estimated left-sided sinusoid are in orange.

The task is to locate the model switches and at each switch to estimate the right-sided and left-sided sinusoids and noise variances. For that purpose, we use a two-sided LSSM signal with $n_f = n_p = 2$,

$$A_f = A_p^{-1} = \begin{bmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{bmatrix}$$

(8.122)

$$s_f = s_p = [1 \ 0]^T,$$

(8.123)

for some given $\omega \in \mathbb{R}_+$ and with unknown $C_p \in \mathbb{R}^{1 \times 2}$, and $C_f \in \mathbb{R}^{1 \times 2}$.

Except the LSSM model, the algorithm is similar to the one described in the previous part. The log-likelihood ratio is plotted in Fig. 8.13 lower plot, red line. The green dashed lines indicate the local maxima (within a certain window and above a given threshold) of $\text{LLR}_k$, which indicates the presence of a model switch. Furthermore, for each model switch that we detect, we also plot a few samples of the estimated right-sided sinusoid (in magenta) and left-sided sinusoid (in orange) on top of the observed signal. Again, the proposed algorithm outputs good performances in detecting and estimating model switches.
Part III

Learning Sparse Signal Decompositions in a State Space World
“Celui qui trouve sans chercher est celui qui a longtemps cherché sans trouver.”

“The one who finds without seeking is the one who has long sought without finding.”

Gaston Bachelard
Chapter 9

Why Should We Learn Sparse Signal Decompositions?

Many signals inherently exhibit similarities across time and across measurement channels. These resemblances are revealed by occurrences of signal shapes with common characteristics. By exploiting temporal and spatial resemblances, we can describe a given signal with a sparse multichannel representation, which can be interpreted as a signal labeling.

A temporal resemblance arises when signal shapes with common characteristics successively and continuously occur across time. For instance, a temporal resemblance is naturally present in a standard ECG recording. In this case, the patient’s heart continuously beats and at each heart beat, a cardiac cycle (depolarization and polarization) creates characteristic electrical signal shapes in the ECG recording. In a standard ECG recording configuration and for most healthy people, the signal shapes produced by all heart beats look alike. Thus, the knowledge of a typical signal shape produced by a heart beat and the occurrences of the heart beat events establish a sparse representation of the whole ECG recording.

On the other hand, a spatial resemblance exhibits signal shapes in each channel with shared properties. Furthermore, the signal shapes in all channels can appear either simultaneously or with some time delays. In order to illustrate spatial resemblance, consider a sensor with sev-
eral electrode points spatially arranged close together one after another. When a moving electric dipole is approaching the sensor, neighboring electrode points output very similar signal shapes in their electric potential signals. Then, by analyzing the voltage signals in all measurement channels, it might be possible to extract the speed and direction of the electric dipole, which thus provides a sparse representation of the signal shapes in all the measurement channels.

In addition, a given signal often exhibits not only one but several types of resemblances. For instance, an abdominal ECG recording of a pregnant woman naturally features two types of resemblances. While all maternal heart beats produce very similar signal shapes of high amplitude, all fetal heart beats also generate their characteristic signal shapes of lower amplitude. In this example, the set of maternal and fetal heart beat events (properly categorized) and the characteristic shapes of the fetal and maternal signals constitute a sparse representation of the ECG recording.

These three examples motivate the need for learning sparse signal decompositions. Ideally, each signal component should account for a characteristic resemblance, which is translated by having a sparse representation.

In Part III, we provide a general approach to learn sparse signal decompositions based on linear state space models. In Chapter 10, we model each signal component with its own, known or unknown LSSM, triggered by sparse inputs. After introducing sparsity with the use of normal variables with unknown variances (cf. Chapter 11), we derive algorithms for learning sparse signal decompositions in Chapter 12. Chapter 13 presents several relevant applications with practical and simulated examples. In particular, we address the problems of learning repetitive signal shapes, learning classes of signal shapes, and decomposing a signal with scaled, time-shifted, and time-dilated versions of a signal shape. Finally, Chapter 14 outlines ideas on how to handle more structures in the sparse representation.
Chapter 10

A State Space World for Sparse Signal Representations

10.1 Measurement and Signal Component Models

Let $M \in \mathbb{N}$. Given $M$-channel discrete-time measurements $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$ (and $K$ is usually large), we wish to explain our signal $y = (y_1, \ldots, y_K)$ as a linear combination of $L \in \mathbb{N}$ vector-valued signal components $s^{[\ell]}_k \in \mathbb{R}^{q_\ell}$, $k \in \{1, \ldots, K\}$, $q_\ell \in \mathbb{N}$, $\ell \in \{1, \ldots, L\}$. Specifically,

$$y_k = D \begin{bmatrix} s^{[1]}_k \\ \vdots \\ s^{[L]}_k \end{bmatrix} + Z_k, \quad (10.1)$$

for $k \in \{1, \ldots, K\}$, with $D \in \mathbb{R}^{M \times Q}$, $Q = \sum_{\ell=1}^L q_\ell$, and $Z_k \sim \mathcal{N}(0, \sigma_Z^2 I_M)$. Additionally, each signal component is assumed to be sparsely represented in the input domain of its own, known or unknown LSSM driven by i.i.d. Gaussian noise. Precisely, $s^{[\ell]}_k \in \mathbb{R}^{q_\ell}$, $k \in \{1, \ldots, K\}$, is explained as the output of a LSSM of a given order $n_\ell \in \mathbb{N}$ and triggered by sparse input vectors $u^{[\ell]}_k \in \mathbb{R}^{j_\ell}$, $j_\ell \in \mathbb{N}_0$, such
that

\[
\begin{align*}
X_k^{[\ell]} & = A_{\ell} X_{k-1}^{[\ell]} + B_k^{[\ell]} u_k^{[\ell]} + E_k^{[\ell]} \\
S_k^{[\ell]} & = C_{\ell} X_k^{[\ell]},
\end{align*}
\tag{10.2}
\]

with states \(X_k^{[\ell]} \in \mathbb{R}^{n_\ell}\) (\(X_0^{[\ell]} = 0\)), \(A_{\ell} \in \mathbb{R}^{n_\ell \times n_\ell}\), \(C_{\ell} \in \mathbb{R}^{q_\ell \times n_\ell}\), \(B_k^{[\ell]} \in \mathbb{R}^{n_\ell \times j_\ell}\), and \(n_\ell\)-dimensional i.i.d. Gaussian state noise \(E_k^{[\ell]} \sim \mathcal{N}(0, V_{E^{[\ell][\ell]}})\) with \(V_{E^{[\ell][\ell]}} \in \mathbb{R}^{n_\ell \times n_\ell}\) a symmetric positive definite matrix or a zero matrix. We denote

\[
\theta_\ell = \{C_{\ell}, A_{\ell}, B_1^{[\ell]}, \ldots, B_K^{[\ell]}, V_{E^{[\ell][\ell]}}\}
\tag{10.3}
\]

the set of LSSM parameters of the \(\ell^{th}\) signal component.

Stacking the time-varying input vectors \(u_k^{[\ell]}\) into a matrix

\[
u^{[\ell]} = \begin{bmatrix} u_1^{[\ell]} & \cdots & u_K^{[\ell]} \end{bmatrix} \in \mathbb{R}^{j_\ell \times K},
\tag{10.4}
\]

sparsity of the inputs means that the number of non-zero elements of the matrix \(u^{[\ell]}\) is substantially smaller than \(j_\ell \cdot K\). In particular, sparsity is considered equally over the time indices \(k \in \{1, \ldots, K\}\) and the input dimensions \(j_\ell \in \{1, \ldots, j_\ell\}\). The sparsity constraint will be introduced in Chapter 11.

All signal components are considered mutually independent; the random variables \(E_k^{[\ell]}, \ell \in \{1, \ldots, L\}\), are statistically independent, there is neither a link between the LSSM parameter sets \(\theta_\ell, \ell \in \{1, \ldots, L\}\) nor between the inputs \(u^{[\ell]}, \ell \in \{1, \ldots, L\}\). This is actually not a restriction since any known dependency is handled by extending either the input dimension \(j_\ell\) or the output dimension \(q_\ell\) of a signal component.

We now summarize the statistical model of the measurements and explore the modeling capabilities of this signal component model.

### 10.2 Joint LSSM of the Measurement Model

Combining (10.1) and (10.2), the discrete-time observations \(y_k \in \mathbb{R}^M, k \in \{1, \ldots, K\}\), are explained as the output of a joint LSSM representation of order \(n = \sum_{\ell=1}^L n_\ell\) and given by

\[
\begin{align*}
X_k &= AX_{k-1} + B_k u_k + E_k \\
y_k &= C X_k + Z_k,
\end{align*}
\tag{10.5}
\]
with
\[ C = D \text{ diag}(C_1, \ldots, C_L) \in \mathbb{R}^{M \times n} \quad (10.6) \]
\[ A = \text{ diag}(A_1, \ldots, A_L) \in \mathbb{R}^{n \times n} \quad (10.7) \]
\[ B_k = \text{ diag}(B_k^{[1]}, \ldots, B_k^{[L]}) \in \mathbb{R}^{n \times J} \quad (10.8) \]
\[ V_E = \text{ diag}(V_E^{[1]}, \ldots, V_E^{[L]}) \in \mathbb{R}^{n \times n}, \quad (10.9) \]
where \( J = \sum_{\ell=1}^L j_\ell \). The states \( X_k \in \mathbb{R}^n \), inputs \( u_k \in \mathbb{R}^J \), random vectors \( E_k \in \mathbb{R}^n \) consist of stacking, respectively, the signal component states \( X_k^{[\ell]} \), inputs \( u_k^{[\ell]} \), and random vectors \( E_k^{[\ell]} \), \( \ell \in \{1, \ldots, L\} \).

Denoting
\[ \theta = \left( \bigcup_{\ell=1}^L \theta_\ell \right) \bigcup \{ \sigma^2_Z \}, \quad (10.10) \]
the set of all LSSM parameters and \( u = (u_1, \ldots, u_K) \), the joint probability density function arising from (10.5) expresses as
\[ p(y, x|u, \theta) = p(x_0) \prod_{k=1}^K p(y_k|x_k, \theta)p(x_k|x_{k-1}, u_k, \theta), \quad (10.11) \]
with
\[ p(y_k|x_k, \theta) = \mathcal{N}(y_k : C x_k, \sigma^2_Z I_M) \quad (10.12) \]
\[ p(x_k|x_{k-1}, u_k, \theta) = \mathcal{N}(x_k : A x_{k-1} + B_k u_k, V_E) \quad (10.13) \]
\[ p(x_0) = \delta(x_0). \quad (10.14) \]

The transition probability density function (10.13) further factorizes for each individual signal component, i.e.,
\[ p(x_k|x_{k-1}, u_k, \theta) = \prod_{\ell=1}^L \mathcal{N}(x_k^{[\ell]} : A_\ell x_{k-1}^{[\ell]} + B_k^{[\ell]} u_k^{[\ell]}, V_{E[\ell]}). \quad (10.15) \]
Each factor in (10.15) only depends on the parameter set \( \theta_\ell \) of a single signal component.

Finally, denoting \( y = \text{ vec}(y) \), \( u = \text{ vec}(u) \), the likelihood function
\[ p(y|u, \theta) = \int p(y, x|u, \theta)dx \quad (10.16) \]
\[ = \mathcal{N}(y : Hu, V_Y), \quad (10.17) \]
with a symmetric positive definite matrix $V_Y \in \mathbb{R}^{MK \times MK}$ and $H \in \mathbb{R}^{MK \times JK}$ both depending on the LSSM parameter $\theta$, is a Gaussian density in $y \in \mathbb{R}^{MK}$ and proportional to a Gaussian density in $u \in \mathbb{R}^{JK}$. For explicit relations between $\theta$, $H$, and $V_Y$, we refer to Chapter 2 and in particular (2.14).

This model is equivalent to explaining $y$ with

$$y = Hu + \tilde{Z}$$

(10.18)

where $\tilde{Z} \sim \mathcal{N}(0, V_Y)$, $u \in \mathbb{R}^{JK}$ is an unknown sparse vector, and $H$ and $V_Y$ are unknown but structured and constrained by the LSSM model (10.5). This equality enlightens how the LSSM strongly regularizes the estimation problem. Note that in standard blind source separation (BSS) literature 20, the elements of $u_k \in \mathbb{R}^J$, $k \in \{1, \ldots, K\}$, are referred to the $J$ source signals. However, dealing directly with the $L$ signal components $s^{[\ell]}$, $\ell \in \{1, \ldots, L\}$ is not restrictive and offers the possibility of having a signal component $s^{[\ell]}$ with no inputs (i.e., $u_k^{[\ell]} = 0$ for all $k$), as described in the next section.

### 10.3 Capabilities of the Signal Component Model

The signal component model in (10.2), which is a Kalman filter model 36, has a simple description and yet is rich enough to model a plethora of signal components encountered in applications (cf. Chapter 13). Its charm also consists in its flexibility to incorporate a priori knowledge by constraining the LSSM parameter set $\theta_\ell$. For instance, a choice of parametrization of the LSSM in (10.2) implies constraints on $\theta_\ell$.

Except for the time-dependent input $u_k^{[\ell]}$, the input matrix $B_k^{[\ell]}$ possibly depends on the time index $k$ and is subject to time-dependent and/or individual constraints. In the following, we focus on simple time-dependent constraints such as $B_k^{[\ell]}$ equal over time or no time-dependent constraint at all. Indeed, more involved time-dependent constraints would require extra hierarchical modeling layers, which is not the main focus of this thesis but nonetheless mentioned in Chapter 14. However, individual (i.e. not time-dependent) constraints can be included by individually restricting $B_k^{[\ell]} \in B_\ell$ where $B_\ell \subset \mathbb{R}^{n_\ell \times j_\ell}$.

We now specialize the general signal component model in (10.2) into cases of interest which are used in Chapter 13. Combinations of those
cases might be possible if needed.

Note that when no state noise (i.e., $V_E[\ell] = 0$) is added to the LSSM, the signal component in (10.2) writes as

$$s_k^{[\ell]} = \sum_{i=1}^{k} C_{\ell} A_{\ell}^{-i} B_i^{[\ell]} u_i^{[\ell]}.$$  \hfill (10.19)

### 10.3.1 Repetitions of a Single Signal Shape

When $j_\ell = 1$ and $B_k^{[\ell]} = B^{[\ell]} \in \mathbb{R}^{n_{\ell}}$ is independent of $k$, the signal component $s^{[\ell]}$ is composed of scaled and time-shifted versions of the signal shape

$$\phi_\ell(i) = \begin{cases} 0, & i < 0 \\ C_{\ell} A_{\ell}^i B^{[\ell]}, & i \geq 0. \end{cases}$$  \hfill (10.20)

When $V_E[\ell] = 0$, we have the equality

$$s_k^{[\ell]} = \sum_{i=1}^{K} \phi_\ell(k - i) u_i^{[\ell]}.$$  \hfill (10.21)

When $V_E[\ell] \neq 0$, (10.21) is valid up to an additive noise term. Note that the addition of state noise can compensate for variations of the signal shape, which will reveal to be useful in practice. Here, sparsity means that the total number of occurrences of the signal shape over time is small compared to $K$.

This model encompasses the convolutive mixture model of [68] when $q_\ell = M$. However, unlike standard convolutive mixture models, our model deals with both finite-length and infinite-length signal shapes within the same representation and with the same complexity. Indeed, a finite-length signal shape would further require $A_{\ell}$ to be nilpotent such that $\phi_\ell(i)$ vanishes once passed a certain index $i$.

### 10.3.2 Repetitions of Several Signal Shapes

When $B_k^{[\ell]} = B^{[\ell]} \in \mathbb{R}^{n_{\ell} \times j_\ell}$ is independent of $k$, the signal component $s^{[\ell]}$ is composed of overlapping scaled and time-shifted versions of $j_\ell$ signal shapes

$$\phi_\ell^{(j)}(i) = \begin{cases} 0, & i < 0 \\ C_{\ell} A_{\ell}^i \{B^{[\ell]}\}_{j_\ell}^i, & i \geq 0. \end{cases}$$  \hfill (10.22)
for $j \in \{1, \ldots, j_{\ell}\}$. When $V_{E[\ell]} = 0$, we have the equality

$$s_{k}^{[\ell]} = \sum_{i=1}^{K} \sum_{j=1}^{j_{\ell}} \phi_{\ell}^{(j)} (k - i) \{ u_{i}^{[\ell]} \}_{j}.$$  \hspace{1cm} (10.23)

When $V_{E[\ell]} \neq 0$, (10.23) is valid up to an additive noise term. Now, sparsity means that the total number of occurrences of the $j_{\ell}$ signal shapes over time is small compared to $j_{\ell} \cdot K$.

**10.3.3 Signal Shapes From a Given Class**

Let $B_{\ell}$ denote a given subset of $\mathbb{R}^{n_{\ell}}$. When $j_{\ell} = 1$ and $B_{k}^{[\ell]} \in B_{\ell}$ is dependent on $k$, the signal component $s^{[\ell]}$ is composed of overlapping signal shapes from a signal class. All signal shapes are generated by a common system of parameter $(C_{\ell}, A_{\ell})$ but with different directions $B_{k}^{[\ell]} \in B_{\ell}$. Specifically, when $V_{E[\ell]} = 0$, we have

$$s_{k}^{[\ell]} = \sum_{i=1}^{k} C_{\ell} A_{\ell}^{k-i} B_{i}^{[\ell]} u_{i}^{[\ell]}.$$  \hspace{1cm} (10.24)

When $V_{E[\ell]} \neq 0$, (10.24) is valid up to an additive noise term. If an input triggers at time index $i$, it produces the signal shape

$$\phi_{\ell}^{(i)}(k) = \begin{cases} 0, & k < i \\ C_{\ell} A_{\ell}^{k-i} B_{i}^{[\ell]}, & k \geq i \end{cases},$$  \hspace{1cm} (10.25)

from the signal class. Here, sparsity means that the number of occurrences of a signal shape from the signal class is small compared to the duration $K$.

Even if we have assumed that $j_{\ell} = 1$, extending this signal component model such that $j_{\ell} \in \mathbb{N}$ where $B_{k}^{[\ell]}$ is constrained to some set $B_{\ell} \in \mathbb{R}^{n_{\ell} \times j_{\ell}}$ is straightforward. The need for such a model can be justified as follows. Consider a LSSM model for a polynomial of degree $n_{\ell} - 1$. Then, $A_{\ell}$ and $C_{\ell}$ can be fixed as explained in Sec. 3.3 and specifying one column of $B_{k}^{[\ell]}$ is equivalent to setting the polynomial coefficients. Now, assume that we add constraints on the columns of $B_{k}^{[\ell]}$ such that its $d^{th}$ column can only create a signal shape that is a polynomial of degree at most $d - 1$. Since sparsity is equally considered over time and input dimension, a sparsity constraint should somehow automatically select the degree of the polynomial. Therefore, the case $j_{\ell} > 1$ is suited for model selection in some particular cases.
10.3.4 Signal Shapes Generated From a System

When $j_\ell = 1$ and $B_k^{[\ell]} \in \mathbb{R}^{n_\ell}$ is actually dependent on $k$, the signal component $s_k^{[\ell]} \in \mathbb{R}^{n_\ell}$ is composed of overlapping signal shapes generated from a common system of parameter $(C_\ell, A_\ell)$ but with different input directions $B_k^{[\ell]}$. This is a special case of “Signal Shapes From a Given Class” (cf. Sec. 10.3.3) where $B_\ell = \mathbb{R}^{n_\ell}$. Thus, sparsity means that the number of times that the system is triggered is small compared to the duration $K$.

A mathematically equivalent model can be achieved using Sec. 10.3.2 where $j_\ell = n_\ell$ and $B_k^{[\ell]} = I_{n_\ell}$ is constant and independent of $k$. Then, the input direction and amplitude is directly included in $u_k^{[\ell]} \in \mathbb{R}^{n_\ell}$, such that

$$s_k^{[\ell]} = \sum_{i=1}^{k} C_\ell A_k^{k-i} u_i^{[\ell]},$$

(10.26)

which can be made equivalent to (10.24). However, the meaning of sparsity differs. While a signal component as in (10.24) is exclusively sparse in time since $u_k^{[\ell]}$ is scalar, a signal component as in (10.26) is simultaneously sparse in time and dimension since $u_k^{[\ell]}$ is a vector (i.e., the number of times that the system is triggered in a specific dimension is small compared to $K \cdot n_\ell$).

10.3.5 Filtered White Gaussian Noise

When the LSSM only has white Gaussian noise inputs (i.e., $u_k^{[\ell]} = 0$ for all $k$), the signal component $s_k^{[\ell]} \in \mathbb{R}^{n_\ell}$ consists of filtered Gaussian noise. Such a signal model can generate smooth signals such as a wandering baseline. Under a suitable choice of parameters, this model can emulate a generating process for spline smoothing [74,75,77]. Note that, counter-intuitively, filtered white Gaussian noise is actually the sparsest signal component in this LSSM representation.
Chapter 11

Sparsity by Gaussian Scale Mixtures and Type-II Estimation

As mentioned in Chapter 10, our model assumes that source signals are sparsely represented in the input domain of a LSSM. Thus, we want to enforce the input vectors to be sparse while maximizing the likelihood (10.17) in order to estimate all unknown parameters. In this section only, referring to (10.5), we look at the observations $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$, as a single measurement vector $y = \text{vec}(y_1, \ldots, y_K) \in \mathbb{R}^{MK}$ and the inputs $u_k \in \mathbb{R}^J$, $k \in \{1, \ldots, K\}$, as a single input vector $u = \text{vec}(u_1, \ldots, u_K) \in \mathbb{R}^N$ with $N = JK$, which are both linked via the likelihood (cf. (10.17))

$$p(y|u, \theta) = \mathcal{N}(y : Hu, V_Y).$$

Recall that $\theta$ denotes the set of all LSSM parameters.

11.1 A Dictionary Learning Problem

In order to enforce sparsity, a first idea which comes to mind is to introduce a sparsity-promoting prior on $u$. In the classical compressed sensing literature [15], we assume that the input vector is a random vector $U \in \mathbb{R}^N$ with components drawn independently and identically from
a sparsity-promoting prior distribution $f(v), v \in \mathbb{R}$, such as a Laplace or a Student’s $t$ distribution, i.e.,

$$p(u) = \prod_{i=1}^{N} f(u_i). \quad (11.2)$$

Then, standard dictionary learning algorithms, such as $K$-SVD \cite{2}, would iteratively alternate between estimating a sparse vector $u$ given the LSSM parameters $\theta$ (with a standard compressed sensing algorithm such as matching pursuit \cite{51}) and updating $\theta$ given a sparse vector $u$ (e.g., \cite{2,42,55}).

This approach meets several obstacles. First, most standard compressed sensing algorithms have a polynomial complexity in $K$ and actually require the computation of $H$ and $V_Y$ in closed-form. Since $V_Y$ and $H$ scale with the square of the number of samples, we absolutely want to avoid manipulating these huge matrices. Secondly, standard dictionary learning algorithms such as $K$-SVD \cite{2} cannot be easily adapted to our LSSM constraints. Third, such an approach can miserably fail when the dictionary is strongly coherent (i.e., high correlation between some of the columns of the dictionary). In our case, our LSSM indeed produces a highly coherent dictionary. For instance, when a source signal is composed of repetitive signal shapes, we prescribe all shifted versions of this shape, often creating a strongly coherent dictionary. Finally, the sparsity-promoting prior on $u$ is independent of the LSSM parameter set $\theta$ and thus jointly estimating both $u$ and $\theta$ might lead to undesirable effects which counteract each other: nothing prevents the estimates from converging towards a nonsensical LSSM with a non-sparse vector $u$. All these hurdles enhance the need for a different approach for enforcing sparsity.

### 11.2 Gaussian Scale Mixtures and Type-II Estimation

Ideally, we are seeking a regularization which tweaks sparse inputs with system parameters together such that the resulting cost function to minimize is well behaved and efficiently optimized. For that purpose, we use Gaussian scale mixtures (GSMs) \cite{4}. Many sparsity-promoting prior dis-
tributions \( f(v) \) can be represented as a GSM [478], i.e., as
\[
f(v) = \int_0^{+\infty} N(v : 0, \sigma^2) \psi(\sigma^2) \, d\sigma^2, \tag{11.3}
\]
where \( \psi(\sigma^2) \in \mathbb{R} \) is not necessarily a probability density distribution nor non-negative [18,78]. Among others, the Laplace, Student’s \( t \) and exponential power family distributions can be represented with a GSM [478]. One could argue that using a prior on \( u \) with such a GSM representation, i.e.,
\[
p(u) = \prod_{i=1}^{N} \int_0^{+\infty} N(u_i : 0, \sigma^2_{U_i}) \psi(\sigma^2_{U_i}) \, d\sigma^2_{U_i}, \tag{11.4}
\]
does not deal with our primary concerns about the joint estimation of the sparse input \( u \) and system parameter \( \theta \). However, this representation opens new ways of performing estimation and in particular, the type-II estimation [6]. Instead of estimating \( u \) directly, type-II methods consist of first estimating the hyperparameters \( \sigma^2_U = (\sigma^2_{U_1}, \ldots, \sigma^2_{U_N}) \) with
\[
\hat{\sigma}^2_U = \arg\max_{\sigma^2_U} p(y|\theta, \sigma^2_U) \prod_{i=1}^{N} \psi(\sigma^2_{U_i}) \tag{11.5}
\]
where
\[
p(y|\theta, \sigma^2_U) = \int p(y|u, \theta) \prod_{i=1}^{N} N(u_i : 0, \sigma^2_{U_i}) \, du_i \tag{11.6}
\]
\[
= N(y : 0, V_Y + H\Sigma_U H^T), \tag{11.7}
\]
where \( \Sigma_U = \text{diag}(\sigma^2_{U_1}, \ldots, \sigma^2_{U_N}) \). Then, \( u \) is estimated as
\[
\hat{u} = \arg\max_{u} p(y|u, \theta) \prod_{i=1}^{N} N(u_i : 0, \hat{\sigma}^2_{U_i})\psi(\hat{\sigma}^2_{U_i}) \tag{11.8}
\]
\[
= \hat{\Sigma}_U H^T(V_Y + H\hat{\Sigma}_U H^T)^{-1}y, \tag{11.9}
\]
which can be seen as performing a MAP estimation on \( u \) with the Gaussian prior distribution
\[
p(u|\hat{\sigma}^2_U) = \prod_{i=1}^{N} N(u_i : 0, \hat{\sigma}^2_{U_i}) \tag{11.10}
\]
\[
= N(u : 0, \hat{\Sigma}_U). \tag{11.11}
\]
The estimate \( \hat{u} \) contains information on the sign of the inputs which is not included in the estimated variances \( \hat{\sigma}_U^2 \). Although (11.9) suggests the computation of \( H, V_Y, \) and a matrix inverse, none of these expensive computations are required since \( \hat{u} \) can be obtained with a Gaussian message passing algorithm \([44, 45]\) as described in Sec. 4.1, which is of complexity \( O(Kn^2) \).

In \([81]\), it is proven that the estimate of \( \sigma_U^2 \) as in (11.5) is sparse. More importantly, even local maxima of \( p(y|\theta, \sigma_U^2) \) are at sparse solutions. As reflected in (11.9), the zero components of \( \hat{\sigma}_U^2 \) also create corresponding zeros in the estimated input \( \hat{u} \); the sparsity structure of \( \hat{\sigma}_U^2 \) is directly transmitted to \( \hat{u} \), and thus preserved. The use of GSM and type-II estimation belongs to the sparse Bayesian learning (SBL) framework \([69, 81, 82]\).

Even if the function \( \psi \) introduced in (11.3) is important to properly define a GSM since the integral in (11.3) must exist, the role of \( \psi \) is actually much less important in the type-II estimation. In fact, as in standard Bayesian modeling, the gain of information obtained by specifying a prior on the hyperparameters decreases with the level of hierarchy \([31, 32]\). Actually, even ignoring \( \psi \) by choosing a uniform improper prior (i.e., \( \psi = 1 \), which can be seen as a limit case of an inverse-gamma prior) does not penalize the estimation, is scale-free, and more importantly still enforces sparsity. This special case is the basis of automatic relevance determination (ARD) \([48, 49, 53, 69, 80]\), which has been used to determine relevant parameters in neural networks \([53]\) and kernel methods \([69]\).

From now on, we always assume that \( \psi = 1 \). Thus, we treat \( \sigma_U^2 \) as a parameter that is estimated by maximum likelihood (cf. (11.7)), i.e.,

\[
J(\theta, \sigma_U^2) = y^T(V_Y + H\Sigma_U H^T)^{-1}y + \ln |V_Y + H\Sigma_U H^T|.
\]

(11.12)

This cost also regularizes the estimation of the LSSM parameters \( \theta \), as explained in the next section.

As proven in \([81\, Theorem 2]\), every local minimum of \( J(\theta, \sigma_U^2) \) with respect to \( \sigma_U^2 \) is achieved at a sparse solution (at most \( KM \) components are non-zero), i.e., many components of \( \hat{\sigma}_U^2 \) are zeros. Essentially, the local minima of (11.12) are also minima of an optimization problem which minimizes a concave function on a convex polytope, which is known to create sparse solutions \([47]\).

In a nutshell, type-II estimation with GSM can be seen as modeling sparse components with zero-mean normal variables with unknown variance (NUV) \([44]\) and estimating the variances along with the un-
known model parameters by maximum likelihood. On the other hand, since \( p(y|u, \theta) \) has a Gaussian graphical model, the use of an expectation maximization (EM) algorithm is well-suited for estimating both the unknown variances and the LSSM parameters. Indeed, all expectation quantities required in the EM update can be computed efficiently with Gaussian message passing. Not only do type-II methods with GSM bring sparsity to Gaussian graphical models and in particular LSSM, they also preserve the Gaussian nature of estimation when using an EM algorithm. Such an approach has been advocated in [11, 44, 85, 87]. We also point out the success of such a framework in reconstructing multi-dimensional signals as demonstrated in [88].

11.3 Regularization and Type-II Estimation

We now investigate the nature of the effective regularization on \( u \) and \( \theta \) which is intrinsically brought by type-II estimation with the use of GSM models. We here follow (and slightly extend) the analysis developed in [80].

First, we define the function

\[
L_\theta(u, \sigma_{U}^2) = (y - Hu)^T V_Y^{-1} (y - Hu) + \sum_{i=1}^{N} \frac{u_i^2}{\sigma_{U_i}^2} + \ln |V_Y + H \Sigma_U H^T|. \tag{11.13}
\]

From the relation

\[
\min_u (y - Hu)^T V_Y^{-1} (y - Hu) + \sum_{i=1}^{N} \frac{u_i^2}{\sigma_{U_i}^2} = y^T (V_Y + H \Sigma_U H^T)^{-1} y, \tag{11.14}
\]

we deduce that

\[
\min_u L_\theta(u, \sigma_{U}^2) = J(\theta, \sigma_{U}^2), \tag{11.15}
\]

and thus, for all \((u, \sigma_{U}^2, \theta)\), we have

\[
L_\theta(u, \sigma_{U}^2) \geq J(\theta, \sigma_{U}^2). \tag{11.16}
\]

Consequently, \( L_\theta(u, \sigma_{U}^2) \) is a tight upper-bounding function of \( J(\theta, \sigma_{U}^2) \). In addition, it results that \( \hat{\sigma}_{U}^2 \) which minimizes \( (11.12) \) satisfies

\[
\hat{\sigma}_{U}^2 = \arg\min_{\sigma_{U}^2} \min_u L_\theta(u, \sigma_{U}^2) \tag{11.17}
\]
and also,

\[ (\hat{\sigma}_U^2, \hat{u}) = \arg\min_{\sigma_U^2, u} \mathcal{L}_\theta(u, \sigma_U^2). \]  

(11.18)

Analogously, \( \hat{u} \) as defined in (11.18) satisfies

\[ \hat{u} = \arg\min_u \min_{\sigma_U^2} \mathcal{L}_\theta(u, \sigma_U^2) \]

(11.19)

\[ = \arg\min_u \mathcal{L}_\theta(u, \hat{\sigma}_U^2), \]

(11.20)

which shows that \( \hat{u} \) coincides with the one defined in (11.9). Furthermore, we can rewrite (11.19) as

\[ \hat{u} = \arg\min_u (y - Hu)^T V_Y^{-1} (y - Hu) + h_\theta^*(u^2), \]

(11.21)

where

\[ h_\theta^*(u^2) = \min_{\sigma_U^2} \sum_{i=1}^N \frac{u_i^2}{\sigma_{U_i}^2} + \ln |V_Y + H \Sigma U H^T| \]

(11.22)

is the concave conjugate of \( h_\theta(\sigma_U^{-2}) = -\ln |V_Y + H \Sigma U H^T| \). Despite its subterfuge, type-II estimation reveals to be quite powerful by essentially changing the nature of the effective prior on \( u \) which becomes, following our analysis in (11.21),

\[ p(u|\theta) \propto e^{-\frac{1}{2} h_\theta^*(u^2)}. \]

(11.23)

Thus, the components of \( U \) are not, in general, independent. Since \( h_\theta^*(u^2) \) is concave and non-decreasing (see [80]), the prior on \( u \) promotes sparsity. Even more importantly, this prior depends on the LSSM parameter \( \theta \) through \( H \) and \( V_Y \), and thus, also regularizes the estimation of \( \theta \). Further properties of the prior distribution in (11.23) have been investigated in [80]. Especially, this prior is scale-invariant with respect to the columns of \( H \). Moreover, it is a tighter approximation of the \( L_0 \) semi-norm compared to the \( L_1 \) norm while producing fewer local minima than using the \( L_0 \) semi-norm [80].

Note also that, type-II estimation also brings a log-determinant term in (11.12), which penalizes trivial LSSMs that neglect correlation in time or within measurement channels. Indeed, as a consequence of Hadamard inequality,

\[ \ln |V_Y + H \Sigma U H^T| \leq \sum_{i=1}^{MK} \ln \left( \{V_Y + H \Sigma U H^T\}_{i,i} \right) \]

(11.24)
and equality holds when $V_Y + H\Sigma_U H^T$ is diagonal, that is to say, when the LSSM does not exhibit correlation in time and within measurement channels. Thus, type-II estimation also regularizes the estimation of the LSSM parameters in a suitable manner.

On the other hand, the cost function (11.12) exhibits two properties that we should be aware of. First, the cost function (11.12) is unaffected by moving a positive scale factor between any column of $H$ and its associated component in $\sigma_U$. When estimating $\theta$, a rescaling of the columns of $H$ in some manner can be performed if needed. Secondly, the cost function (11.12) is unaffected by a change of sign of any column of $H$. Without a priori knowledge, this information cannot be recovered. These two invariance properties are in common with standard blind source separation settings [20].

11.4 A Structured Covariance Estimation Problem

We here further investigate the cost function $J(\theta, \sigma_U^2)$ in (11.12), which we recall to have the expression

$$J(\theta, \sigma_U^2) = y^T (V_Y + H\Sigma_U H^T)^{-1} y + \ln |V_Y + H\Sigma_U H^T|.$$  \hspace{1cm} (11.25)

For the exact expression of $V_Y$, we refer to (2.21).

By defining the covariance matrix

$$V_{\text{tot}} = V_Y + H\Sigma_U H^T,$$ \hspace{1cm} (11.26)

minimizing $J(\theta, \sigma_U^2)$ is equivalent to minimizing

$$\tilde{J}(V_{\text{tot}}) = \text{tr} \left( V_{\text{tot}}^{-1} yy^T \right) + \ln |V_{\text{tot}}|,$$ \hspace{1cm} (11.27)

where $V_{\text{tot}}$ is structured by the LSSM. Thus, estimating $\theta$ and $\sigma_U^2$ consist of a structured covariance estimation problem (e.g., see [14]). Actually, this estimation problem can be seen as a constrained, extended version of a probabilistic PCA problem [64,70] or factor analysis, as described in Appendix E.

Note that the function $\tilde{J}(V_{\text{tot}})$ is convex in $V_{\text{tot}}^{-1}$ but in general not in $V_{\text{tot}}$. 
11.4.1 Characterization of Local Minima

The differential of $\tilde{J}(V_{\text{tot}})$ is given by

$$d\tilde{J} = \text{tr} \left( \left( V_{\text{tot}}^{-1} - V_{\text{tot}}^{-1}yy^TV_{\text{tot}}^{-1} \right) (dV_{\text{tot}}) \right),$$  \hspace{1cm} (11.28)

where we have used (A.8) and (A.9). Unfortunately, it is not easy to characterize $dV_{\text{tot}}$, which represents the (infinitesimal) variation of $V_{\text{tot}}$ when changing both the LSSM parameters $\theta$ and the input variances $\sigma_U^2$.

Nevertheless, focusing on the partial derivative with respect to a single input parameter $\sigma_{U_i}$, we have

$$\frac{\partial V_{\text{tot}}}{\partial \sigma_{U_i}} = 2\sigma_{U_i} h_i h_i^T,$$  \hspace{1cm} (11.29)

where $h_i \in \mathbb{R}^{MK}$ denotes the $i^{\text{th}}$ column of $H$, since

$$H\Sigma_U H^T = \sum_{i=1}^{N} \sigma_{U_i}^2 h_i h_i^T.$$  \hspace{1cm} (11.30)

Thus, we obtain

$$\frac{\partial J}{\partial \sigma_{U_i}} = 2\sigma_{U_i} \text{tr} \left( \left( V_{\text{tot}}^{-1} - V_{\text{tot}}^{-1}yy^TV_{\text{tot}}^{-1} \right) h_i h_i^T \right)$$

$$= 2\sigma_{U_i} \left( h_i^TV_{\text{tot}}^{-1}h_i - (h_i^TV_{\text{tot}}^{-1}y)^2 \right).$$  \hspace{1cm} (11.31)

Setting (11.32) to zero, we get

$$\sigma_{U_i} = 0$$  \hspace{1cm} (11.33)

or

$$\left( h_i^TV_{\text{tot}}^{-1}y \right)^2 = h_i^TV_{\text{tot}}^{-1}h_i.$$  \hspace{1cm} (11.34)

An alternative expression of this equality is given in (12.83), which is obtained from maximizing the marginal likelihood with respect to a single parameter $\sigma_{U_i}$. But still, $V_{\text{tot}}$ depends on $\sigma_{U_i}^2$. Expanding $V_{\text{tot}}$ as

$$V_{\text{tot}} = V_{\text{tot}}^{(i)} + \sigma_{U_i}^2 h_i h_i^T$$  \hspace{1cm} (11.35)

with

$$V_{\text{tot}}^{(i)} = V_Y + \sum_{j \neq i} \sigma_{U_j}^2 h_j h_j^T$$  \hspace{1cm} (11.36)

and

$$W_{\text{tot}}^{(i)} = \left( V_{\text{tot}}^{(i)} \right)^{-1}.$$  \hspace{1cm} (11.37)
independent of $\sigma_{U_i}$, the equality in (11.34) rewrites as
\[
\sigma_{U_i}^2 = \frac{(h_i^T W_{tot}^{(i)} y)^2 - h_i^T W_{tot}^{(i)} h_i}{(h_i^T W_{tot}^{(i)} h_i)^2},
\]
where we have used the matrix inversion lemma (A.4) to invert $V_{tot}$:
\[
V_{tot}^{-1} = W_{tot}^{(i)} - W_{tot}^{(i)} h_i \frac{\sigma_{U_i}^2}{1 + \sigma_{U_i}^2 h_i^T W_{tot}^{(i)} h_i} h_i^T W_{tot}^{(i)}.
\]

Thus, in a nutshell, setting the gradient (11.32) to zero leads to
\[
\sigma_{U_i}^2 = \max \left(0, \frac{(h_i^T W_{tot}^{(i)} y)^2 - h_i^T W_{tot}^{(i)} h_i}{(h_i^T W_{tot}^{(i)} h_i)^2}\right),
\]
which also expresses as (12.72) when optimizing the marginal likelihood. Unfortunately, $W_{tot}^{(i)}$ still depends on $\theta$ and also all the remaining $\sigma_{U_j}^2$, for $j \neq i$. Thus, the minimization of (11.12) is usually done iteratively.

Another point worth mentioning is that the local minima of (11.12) are actually roots of some multivariate polynomials in $\sigma_{U_i}^2$. Indeed, since the determinant of a matrix is a polynomial expression of the matrix entries, we have
\[
|V_Y + H \Sigma_U H^T| = Q_\theta(\sigma_{U_i}^2),
\]
where $Q_\theta(\sigma_{U_i}^2)$ is a multivariate polynomial in $\sigma_{U_i}^2$. Furthermore, since
\[
V_{tot}^{-1} = \frac{1}{|V_{tot}|} \text{adj}(V_{tot}),
\]
where $\text{adj}(V_{tot})$ is the adjugate matrix of $V_{tot}$ (i.e., the transpose of the cofactor matrix) and since each cofactor is a determinant, it follows that
\[
y^T V_{tot}^{-1} y = P_\theta(\sigma_{U_i}^2) \frac{Q_\theta(\sigma_{U_i}^2)}{Q_\theta(\sigma_{U_i}^2)},
\]
where $P_\theta(\sigma_{U_i}^2)$ is a multivariate polynomial in $\sigma_{U_i}^2$. As a result, we have
\[
J(\theta, \sigma_{U_i}^2) = \frac{P_\theta(\sigma_{U_i}^2)}{Q_\theta(\sigma_{U_i}^2)} + \ln Q_\theta(\sigma_{U_i}^2),
\]
and
\[
\frac{\partial J}{\partial \sigma_{U_i}} = \frac{1}{Q_\theta(\sigma_{U_i}^2)} \left( \frac{\partial P_\theta}{\partial \sigma_{U_i}} Q_\theta(\sigma_{U_i}^2) + (Q_\theta(\sigma_{U_i}^2) - P_\theta(\sigma_{U_i}^2)) \frac{\partial Q_\theta}{\partial \sigma_{U_i}} \right).
\]
Thus, the local minima of (11.12) are actually roots of some multivariate polynomials in $\sigma_{U_i}$. 
11.4.2 Characterization of $V_{\text{tot}}$

Unknown $\sigma^2_Z$

Let $\Theta$ denote the set of all $\theta$’s that we allow by prescribing a given structured and constrained LSSM. Thus, $V_{\text{tot}}$, as defined in (11.26) and (2.21), belongs to the set

$$V = \{ \sigma^2_Z I_{MK} + PV_E P^T + H\Sigma_U H^T : \sigma^2_U \in \mathbb{R}_+^N, \theta \in \Theta \}. \tag{11.46}$$

It is straightforward to prove that $V$ is a cone if $\sigma^2_Z$ is unconstrained and all positive scaling of the state noise covariance is allowed. However, $V$ is not a convex cone. In any case, $V \subset \mathbb{S}_+$. The way we design our LSSM should not allow $V = \mathbb{S}_+$. Otherwise, the LSSM does not regularize the covariance estimation problem anymore and leads to an unsatisfactory estimation of the parameters such that

$$\hat{V}_{\text{tot}} = yy^T, \tag{11.47}$$

with $\tilde{J}(\hat{V}_{\text{tot}}) \to -\infty$. The estimated covariance matrix simply reproduces the sample covariance matrix $yy^T$.

Fixed $\sigma^2_Z$

When $\sigma^2_Z$ is fixed, $V$ is an affine cone since

$$V = \sigma^2_Z I_{MK} + \tilde{V}, \tag{11.48}$$

with

$$\tilde{V} = \{ PV_E P^T + H\Sigma_U H^T : \sigma^2_U \in \mathbb{R}_+^N, \theta \in \Theta \}. \tag{11.49}$$

It is still not desirable to have a LSSM structure such that $\tilde{V} = \mathbb{S}_+$. Indeed, in such a case, the solution to the minimization of (11.27) leads to

$$\hat{V}_{\text{tot}} = \sigma^2_Z I_{MK} + \frac{||y||^2 - \sigma^2_Z}{||y||^2} yy^T, \tag{11.50}$$

and

$$\tilde{J}(\hat{V}_{\text{tot}}) = 1 + (KM - 1) \ln \sigma^2_Z + \ln ||y||^2. \tag{11.51}$$
The proof is similar to the one derived in Appendix E.2. But still, the obtained covariance matrix estimate is still undesirable since it essentially reproduces the sample covariance \(yy^T\) with a slight regularization. However, for a fixed \(\sigma^2_Z\), we obtain the inequality

\[
\min_{\theta \in \Theta, \sigma^2_U} J(\theta, \sigma^2_U) = \min_{V_{\text{tot}} \in \mathcal{V}} J(V_{\text{tot}}) \geq 1 + (KM - 1) \ln \sigma^2_Z + \ln \|y\|^2, \tag{11.52}
\]

which is a lower bound for the constrained minimization of (11.27).

### 11.4.3 Simultaneous Diagonalization

Since \(V_{\text{tot}}\) is a symmetric positive definite matrix and \(yy^T\) a symmetric positive semi-definite matrix of rank one, there exists a transform matrix \(T\) such that both \(V_{\text{tot}}\) and \(yy^T\) are diagonal and with non-negative diagonal entries, i.e.,

\[
T^T V_{\text{tot}} T = \Lambda = \text{diag}(\lambda) \tag{11.53}
\]
\[
T^T yy^T T = \Gamma = \text{diag}(\gamma), \tag{11.54}
\]

with all \(\lambda_i > 0\) and all \(\gamma_i \geq 0\). The proof of this theorem is given in \[33\]. Furthermore, \(T\) can be chosen to satisfy \(|T| = 1\) by moving a scale factor to the diagonal entries of \(\Lambda\) and \(\Gamma\). Then, we have

\[
|V_{\text{tot}}| = |\Lambda| = \prod_i \lambda_i. \tag{11.55}
\]

Rewriting (11.54), we have

\[
yy^T = (T^{-1})^T \Gamma T^{-1} = \sum_i \gamma_i \tilde{t}_i \tilde{t}_i^T, \tag{11.56}
\]

where \(\tilde{t}_i\) is the \(i^{th}\) column of \((T^{-1})^T\). The only way for (11.56) to hold is that there is an index \(i\) such that \(\tilde{t}_i \propto y\). Furthermore, since \((T^{-1})^T\) is an invertible matrix, this index has to be unique and can be chosen to be \(i = 1\) without loss of generality. Thus, \(\gamma_1 > 0\) (assuming that \(\|y\|^2 \neq 0\)) and \(\gamma_i = 0\), for \(i > 1\). Finally, plugging (11.53) and (11.54) in (11.27), we get

\[
J(\theta, \sigma^2_U) = \frac{\gamma_1}{\lambda_1} + \ln(\lambda_1) + \sum_{i>1} \ln(\lambda_i). \tag{11.57}
\]

Certainly, the relation between \((\theta, \sigma^2_U)\) and \((\gamma_1, \lambda)\) is not explicit and many constraints naturally occurring on \((\gamma_1, \lambda)\) are hard to handle.
Nonetheless, looking at (11.57), maximizing $J(\theta, \sigma_U^2)$ leads to forcing many $\lambda_i, i > 1$ to be as close as possible to zero except for $\lambda_1$ that needs to be close to $\gamma_1$. 
Chapter 12

Learning Sparse Signal Decompositions

We here combine the general measurement model (10.5) with the NUV prior introduced in Chapter 11.

12.1 Maximum Likelihood Estimation

By assumption, each signal component model (10.2) for \( \ell \in \{1, \ldots, L\} \) is triggered by sparse vectors \( U_k^{[\ell]} \in \mathbb{R}^{j\ell} \), which we model as independent zero-mean Gaussian random vectors

\[
U_k^{[\ell]} \sim \mathcal{N} \left( 0, \Sigma U_k^{[\ell]} \right), \tag{12.1}
\]

with diagonal covariance matrices

\[
\Sigma U_k^{[\ell]} = \text{diag} \left( \sigma^2 U_k^{[\ell]} \right) \in \mathbb{R}^{j\ell \times j\ell}, \tag{12.2}
\]

with unknown non-negative diagonal elements \( \sigma^2 U_k^{[\ell]} \in \mathbb{R}^{j\ell} \).

Referring to (10.5), this is equivalent to modeling each input vector component \( U_k^{(j)}, k \in \{1, \ldots, K\}, j \in \{1, \ldots, J\} \), as independent zero-mean Gaussian random variables

\[
U_k^{(j)} \sim \mathcal{N} \left( 0, \sigma^2 U_k^{(j)} \right), \tag{12.3}
\]

145
with unknown variance $\sigma_{U_k}^2 \in \mathbb{R}_+$. Thus, we have

$$U_k \sim \mathcal{N}(0, \Sigma_{U_k}), \quad (12.4)$$

with diagonal covariance matrix

$$\Sigma_{U_k} = \text{diag}(\sigma_{U_k}^2), \quad (12.5)$$

with unknown diagonal elements

$$\sigma_{U_k}^2 = \left(\sigma_{U_{k(1)}}^2, \ldots, \sigma_{U_{k(J)}}^2\right). \quad (12.6)$$

The set of all input parameters is denoted

$$\sigma_U^2 = \bigcup_{k=1}^K \{\sigma_{U_k}^2\}, \quad (12.7)$$

and comprises all the $K \cdot J$ variances $\sigma_{U_k}^2$, $k \in \{1, \ldots, K\}$, $j \in \{1, \ldots, J\}$.

Finding a sparse signal decomposition boils down to estimating $\theta$ and $\sigma_U^2$ by maximizing the likelihood function

$$p(y|\theta, \sigma_U^2) = \int \int p(y, x|u, \theta) p(u|\sigma_U^2) \, du \, dx, \quad (12.8)$$

where $p(y, x|u, \theta)$ is defined in (10.11) and $p(u|\sigma_U^2)$ in (12.3). Recall that $\theta$ is defined in (10.10) and denotes all the LSSM parameters. Note that maximizing (12.8) is equivalent to minimizing the cost function (11.12).

As explained in Chapter 11 at a local maximum of the likelihood function, many $\sigma_{U_k}^2$, $k \in \{1, \ldots, K\}$, $j \in \{1, \ldots, J\}$, are zero. However, among all the parameters in $\theta$, the estimation of the noise variance $\sigma_Z^2$ needs to be treated with care. As a matter of fact, $\sigma_Z^2 = 0$ (perfect data fit) makes the likelihood function (12.8) reach infinity and is unavoidably a global optimum provided that the matrix $V_{\text{tot}} = V_Y + H\Sigma_U H^T$ (cf. (11.12)) is singular. Thus, when jointly performing sparse input estimation and system identification, the estimation of $\sigma_Z^2$ is often compromised and interferes with the sparsity level, as also pointed out in Appendix E.

Consequently, to deal with the trade-off data fit versus sparsity level, we always fix the variance $\sigma_Z^2$ (i.e., the expected data fit), which in turn sets a given range of sparsity level.

For given values of $\theta$ and $\sigma_U^2$, the joint density $p(y, x|\theta, \sigma_U^2)$ factorizes and has a factor graph representation as in Fig. 12.1 (which is the
same as Fig. 4.1). In particular, given values of $\theta$ and $\sigma^2_U$, the posterior estimate of the $\ell$th signal component $\hat{s}^{[\ell]}$ is obtained with

$$\hat{s}^{[\ell]}_k = C^\ell m_{X^{[\ell]}}$$

(12.9)

where $m_{X^{[\ell]}}$ is the posterior mean of $X^{[\ell]}$ and which corresponds to the $\ell$th block of the posterior mean $m_{X_k}$ of $X_k$ (decomposing the column vector in successive blocks of length $n_1, \ldots, n_L$). The computation of $m_{X_k}$ can be done efficiently using Gaussian message passing as described in Sec. 4.1.

We now derive iterative algorithms to maximize (12.8) with respect to both $\theta$ and $\sigma^2_U$.

### 12.2 An EM Algorithm

Maximizing (12.8) has no general closed-form solution. However, the integral form in (12.8) strongly suggests the use of an expectation maximization (EM) algorithm [24]. Moreover, since we have a linear Gaussian model, all expectation quantities required for EM are efficiently computed using Gaussian message passing, as described in Sec. 4.1.

Using $X = (X_1, \ldots, X_K)$ and $U = (U_1, \ldots, U_K)$ as hidden variables
and starting from an initial guess for $\theta$ and $\sigma_U^2$, the EM algorithm iteratively updates $\hat{\theta}$ and $\hat{\sigma}_U^2$ with

$$
(\hat{\theta}, \hat{\sigma}_U^2) = \arg\max_{\theta, \sigma_U^2} \mathbb{E} \left[ \ln \left( p(y, X | U, \theta) p(U | \sigma_U^2) \right) \right], \quad (12.10)
$$

where the expectation is taken with respect to the joint posterior density

$$
p(x, u | y, \theta_{\text{old}}, (\sigma_U^2)_{\text{old}}) \propto p(y, x | u, \theta_{\text{old}}) p(u | (\sigma_U^2)_{\text{old}}). \quad (12.11)
$$

In order to make the EM step in (12.10) well defined, no degenerate Gaussian distributions (i.e., Dirac delta distributions in some dimensions) are allowed within the logarithmic function.

At each EM step, the likelihood (12.8) is guaranteed to be non-decreasing. Thus, the EM estimates converge to a local maximum or a saddle point of the likelihood. Nonetheless, a local maximum is often good enough, as it is the case in the well-known problem of learning a Gaussian mixture. Even if the EM algorithm has the advantage of being numerically stable, it is sensitive to initialization and several restarts might be needed. As far as the convergence rate is concerned, it depends on the fraction between the complete and incomplete Fisher information, as explained in [24]. We only retain that convergence of EM can be slow.

By the virtue of the factorization of the joint distribution in (10.11), the maximization in (12.10) splits for $(C, \sigma_Z^2)$, each $\sigma_{U_k}^2$, $k \in \{1, \ldots, K\}$, $j \in \{1, \ldots, J\}$, and each $\{A_\ell, B_1^{[\ell]}, \ldots, B_K^{[\ell]}, V_{E[\ell]}\}$, $\ell \in \{1, \ldots, L\}$, since

$$
\mathbb{E} [\ln p(y, X, U | \theta)]
= \sum_{k=1}^K \mathbb{E} [\ln \mathcal{N}(y_k : CX_k, \sigma_Z^2 I_M)]
+ \sum_{\ell=1}^L \sum_{k=1}^K \mathbb{E} [\ln \mathcal{N}(X_k^{[\ell]} : A_\ell X_k^{[\ell]} + B_k^{[\ell]} U_k^{[\ell]}, V_{E[\ell]})]
+ \sum_{k=1}^K \sum_{j=1}^J \mathbb{E} \left[ \ln \mathcal{N} \left( U_k^{(j)} : 0, \sigma_U^{2(j)} \right) \right]. \quad (12.12)
$$

Interestingly, all those parameters can be jointly updated. For the
following derivations, it is useful to define

\[
E_{X,X} = \sum_{k=1}^{K} \mathbb{E} \left[ X_k X_k^T \right]
\]  
(12.13)

\[
E_{X\cdot,X} = \sum_{k=1}^{K} \mathbb{E} \left[ X_{k-1} X_k^T \right]
\]  
(12.14)

\[
E_{X\cdot,X\cdot} = \sum_{k=1}^{K} \mathbb{E} \left[ X_{k-1} X_{k-1}^T \right]
\]  
(12.15)

\[
E_{X\cdot,U} = \sum_{k=1}^{K} \mathbb{E} \left[ X_{k-1} U_k^T \right]
\]  
(12.16)

\[
E_{X,U} = \sum_{k=1}^{K} \mathbb{E} \left[ X_k U_k^T \right]
\]  
(12.17)

\[
E_{U,U} = \sum_{k=1}^{K} \mathbb{E} \left[ U_k U_k^T \right]
\]  
(12.18)

\[
E_{X,y} = \sum_{k=1}^{K} \mathbb{E} \left[ X_k \right] y_k^T.
\]  
(12.19)

In accordance with the block structure of the signal component states and inputs (i.e., the blocks \(X_{k}^{[\ell]}\) and \(U_{k}^{[\ell]}\)), the \(\ell\)th diagonal block \(E_{\bullet}^{[\ell]}\) of the quantity \(E_{\bullet}\) involves an expectation over the variables \(X_{k}^{[\ell]}\) and \(U_{k}^{[\ell]}\), \(k \in \{1, \ldots, K\}\) only, such that adding a superscript \((\bullet)^{[\ell]}\) in (12.13)–(12.18) to all quantities makes perfectly sense.

Thus, the input variance EM update is

\[
\hat{\sigma}_{U_{k}^{(j)}}^2 = \mathbb{E} \left[ \left( U_{k}^{(j)} \right)^2 \right],
\]  
(12.20)

for \(k \in \{1, \ldots, K\}\), \(j \in \{1, \ldots, J\}\). Since the local maxima of the likelihood function (12.8) occur when \(\hat{\sigma}_{U_{k}^{(j)}}^2\), \(k \in \{1, \ldots, K\}\), \(j \in \{1, \ldots, J\}\), has only a few non-zero elements, the EM update (12.20) will gradually decrease the value of many \(\hat{\sigma}_{U_{k}^{(j)}}^2\) towards zero. However, this EM update will rarely create exact zero. For that purpose, a marginal likelihood update as described in Sec. 12.4 can be used to create exact zeros when the EM algorithm has almost converged or on a periodic basis. Note that if \(\hat{\sigma}_{U_{k}^{(j)}}^2\) once reaches a zero value, its value will stay zero, as an immediate
consequence of (4.42). Usually, it is beneficial to initialize the input variances to a very small value like $10^{-12}$ such that, ideally, many of them would stay close to zero while only few of them would rise if necessary.

Concerning $(C, \sigma_Z^2)$, the update is

$$\hat{C} = \arg\min_C \sum_{k=1}^K \mathbb{E} \left[ \|y_k - CX_k\|^2 \right]$$

(12.21)

$$\hat{C} = \arg\min_C \text{tr}(C\mathcal{E}_X,C^T - 2C\mathcal{E}_X,y)$$

(12.22)

$$\hat{\sigma}_Z^2 = \frac{1}{KM} \sum_{k=1}^K \mathbb{E} \left[ \|y_k - \hat{C}X_k\|^2 \right].$$

(12.23)

Note that the minimization problem in (12.22) consists in minimizing a quadratic form but usually under constraint. For instance, the matrices $C_\ell, \ell \in \{1, \ldots, L\}$, can often be assumed fixed (since the $B_\ell \{\ell\}$ are updated anyway). Then, following (10.6), we have $C = D\tilde{C}$ where $D$ is the unknown mixing matrix and $C$ is fixed. It follows that

$$\hat{C} = \mathcal{E}^T_{X,y}\tilde{C}^T(\tilde{C}\mathcal{E}_X,C^T)^{-1}\tilde{C}.$$

(12.24)

As mentioned in Sec. 12.1, the update of $\sigma_Z^2$ in (12.23) is rarely used since it interferes with the desired sparsity level; a better data fit (i.e., smaller value of $\hat{\sigma}_Z^2$) commonly requires a less sparse $\hat{\sigma}_U^2$ and a less realistic estimated system.

Concerning the update of $\{A_\ell, B_1^{[\ell]}, \ldots, B_K^{[\ell]}, V_E^{[\ell]}\}$, we need to maximize

$$\sum_{k=1}^K \mathbb{E} \left[ \ln \mathcal{N} \left( X_k^{[\ell]} : A_\ell X_{k-1}^{[\ell]} + B_k^{[\ell]} U_k^{[\ell]}, V_E^{[\ell]} \right) \right],$$

(12.25)

for each $\ell \in \{1, \ldots, L\}$. Note that the maximization problem in (12.25) is well defined only if $V_E^{[\ell]}$ has full rank. We now specialize the maximization according to the different signal component models of Sec. 10.3

While $\{A_\ell, B_1^{[\ell]}, \ldots, B_K^{[\ell]}\}$ can be randomly initialized while satisfying their respective constraints, initializing $V_E^{[\ell]}$ with not too small values is advantageous to favor big enough changes of other system parameters (in particular $A_\ell$). On the other hand, if $V_E^{[\ell]}$ is initialized with too big values, the estimated state trajectory would be perfectly matched with state noise terms only, which would restrain the apparition of non-zero input variances $\hat{\sigma}_U^2$ to compensate for trajectory changes. Thus, the initial value of $V_E^{[\ell]}$ needs to be selected with care.
12.2.1 Repetitions of (One or Several) Signal Shapes

Following Sec. 10.3.1 and 10.3.2, the parameters are such that \( j_\ell \in \mathbb{N} \) and \( B_\ell = B[\ell] \in \mathbb{R}^{n_x \times j_\ell} \) is independent of \( k \). Furthermore, we assume that \( V_{E[\ell]} = \sigma_{E[\ell]}^2 I_{n_\ell} \) (isotropic state noise) where \( \sigma_{E[\ell]}^2 > 0 \) also needs to be estimated. If the signal shapes are well approximated with this LSSM, the estimate of \( \sigma_{E[\ell]}^2 \) should be of small value.

The maximization of (12.25) leads to

\[
\hat{\sigma}_{E[\ell]}^2 = \frac{1}{Kn_\ell} \sum_{k=1}^{K} \mathbb{E} \left[ \| X_k^{[\ell]} - \hat{A}_\ell X_{k-1}^{[\ell]} - \hat{B}_\ell U_k^{[\ell]} \|^2 \right], \tag{12.26}
\]

where \( \hat{A}_\ell \) and \( \hat{B}_\ell \) minimizes

\[
\sum_{k=1}^{K} \mathbb{E} \left[ \| X_k^{[\ell]} - A_\ell X_{k-1}^{[\ell]} - B_\ell U_k^{[\ell]} \|^2 \right] = \text{tr} \left( A_\ell \mathcal{E}_{X,-X}^{[\ell]} A_\ell^T - 2A_\ell \mathcal{E}_{X,-X}^{[\ell]} + \mathcal{E}_{X,X}^{[\ell]} \right)
- 2(\mathcal{E}_{X,U}^{[\ell]} - \hat{A}_\ell \mathcal{E}_{X,-U}^{[\ell]})(B_\ell^T) + B_\ell \mathcal{E}_{U,U}^{[\ell]}(B_\ell^T), \tag{12.27}
\]

where we have expanded the sum of the expected norms and used the equality \( u^T v = \text{tr} (vu^T) \), for any column vectors \( u \) and \( v \) of the same size. Thus, we have

\[
\hat{B}_\ell = \arg\min_{B[\ell] \in \mathcal{B}_\ell} \text{tr} \left( B_\ell \mathcal{E}_{X,-X}^{[\ell]} (B_\ell^T) - 2(\mathcal{E}_{X,U}^{[\ell]} - \hat{A}_\ell \mathcal{E}_{X,-U}^{[\ell]})(B_\ell^T) \right). \tag{12.28}
\]

Assuming an unconstrained \( B[\ell] \) (i.e., \( \mathcal{B}_\ell = \mathbb{R}^{n_x \times j_\ell} \)), we have

\[
\hat{B}_\ell = (\mathcal{E}_{X,U}^{[\ell]} - \hat{A}_\ell \mathcal{E}_{X,-U}^{[\ell]})(\mathcal{E}_{U,U}^{[\ell]} \mathcal{E}_{X,X}^{[\ell]} \mathcal{E}_{X,-U}^{[\ell]} \mathcal{E}_{X,U}^{[\ell]} \mathcal{E}_{U,U}^{[\ell]} - 1), \tag{12.29}
\]

with

\[
\hat{A}_\ell = \arg\min_{A_\ell} \text{tr} \left( A_\ell W_\ell A_\ell^T - 2A_\ell \xi_\ell \right), \tag{12.30}
\]

where

\[
W_\ell = \mathcal{E}_{X,-X}^{[\ell]} - \mathcal{E}_{X,-X}^{[\ell]} (\mathcal{E}_{U,U}^{[\ell]} \mathcal{E}_{X,X}^{[\ell]} \mathcal{E}_{X,-U}^{[\ell]} \mathcal{E}_{X,U}^{[\ell]} \mathcal{E}_{U,U}^{[\ell]} - 1) \mathcal{E}_{X,U}^{[\ell]}, \tag{12.31}
\]

\[
\xi_\ell = \mathcal{E}_{X,-X}^{[\ell]} - \mathcal{E}_{X,-X}^{[\ell]} (\mathcal{E}_{U,U}^{[\ell]} \mathcal{E}_{X,X}^{[\ell]} \mathcal{E}_{X,-U}^{[\ell]} \mathcal{E}_{X,U}^{[\ell]} \mathcal{E}_{U,U}^{[\ell]} - 1) \mathcal{E}_{X,U}^{[\ell]}, \tag{12.32}
\]

which we obtain by plugging (12.29) in (12.27). The minimization problem in (12.30) consists of minimizing a quadratic form in \( A_\ell \) and has...
closed-form solutions when $A_\ell$ is in observable or controllable or Jordan canonical form (cf. Appendix D).

If $A_\ell$ is known, the EM updates for $B_\ell$ and $\sigma^2_{E[\ell]}$ are still given by (12.29) and (12.26) by replacing $\hat{A}_\ell$ by its known value. In addition, further constraints on $B_\ell$ are easily included by solving (12.28) with $B_\ell \subset \mathbb{R}^{n_\ell \times j_\ell}$.

Constraints on $B_\ell$ can also be included in the case of $A_\ell$ unknown but at the price of potentially giving up the closed-form expression for $A_\ell$. If the joint EM update cannot be performed in closed form, it is however possible to alternate EM updates where either $A_\ell$ or $B_\ell$ is updated at a time (but not both at the same time, to ensure a non-decreasing likelihood). However, in the particular case where the constraints still lead to an expression for $\hat{B}_\ell$ that is linear in $\hat{A}_\ell$ and of the form

$$\hat{B}_\ell = X_\ell - \hat{A}_\ell Y_\ell,$$

(12.33)

with $X_\ell, Y_\ell \in \mathbb{R}^{n_\ell \times j_\ell}$ independent of $A_\ell$, the minimization problem over $A_\ell$ remains of the same nature as in (12.30) but with the modified parameters

$$W_\ell = E_{X_\ell, X_\ell} - 2E_{X_\ell, U} Y_\ell^T + Y_\ell E_{U, U} Y_\ell^T$$

(12.34)

$$\xi_\ell = E_{X_\ell, X_\ell} - E_{X_\ell, U} X_\ell^T - Y_\ell (E_{X_\ell, U})^T + Y_\ell E_{U, U} X_\ell^T.$$  

(12.35)

For instance, when $B_\ell$ is known, $\hat{B}_\ell$ is indeed of the form (12.33) with $Y_\ell = 0$ and fixed $X_\ell = B_\ell$. Another example consists of linear constraints of the form

$$B_\ell = t_\ell + b_\ell^T T_\ell,$$

(12.36)

for some fixed matrices $T_\ell, t_\ell$ and unknown, unconstrained matrix $b_\ell$ of suitable dimensions. Then, we find that

$$\hat{B}_\ell = t_\ell + (E_{X_\ell, U} - t_\ell E_{U, U} - \hat{A}_\ell E_{X_\ell, U}) T_\ell^T (T_\ell E_{U, U} T_\ell^T)^{-1} T_\ell,$$

(12.37)

which is of the form (12.33). In particular, this type of constraint handles cases where some of the columns of $B_\ell$ are known while the remaining ones are unknown.

### 12.2.2 Signal Shapes From a Given Class or Generated From a System

Following Sec. 10.3.3 and 10.3.4, the parameters are such that $j_\ell = 1$ and $B_\ell^k \in B_\ell \subset \mathbb{R}^{n_\ell}$ is actually dependent on $k$. Furthermore, we assume
that \( V_{E[\ell]} = \sigma_{E[\ell]}^2 I_{n_\ell} \) (isotropic state noise) where \( \sigma_{E[\ell]}^2 > 0 \) also needs to be estimated.

The maximization of (12.25) leads to

\[
\hat{\sigma}_{E[\ell]}^2 = \frac{1}{K n_\ell} \sum_{k=1}^{K} \mathbb{E} \left[ \| X^\ell_k - \hat{A}_\ell X^\ell_{k-1} - \hat{B}_k^\ell U^\ell_k \|^2 \right],
\]

(12.38)

where \( \hat{A}_\ell \) and \( \hat{B}_k^\ell \), \( k \in \{1, \ldots, K\} \) minimizes

\[
\sum_{k=1}^{K} \mathbb{E} \left[ \| X^\ell_k - A_\ell X^\ell_{k-1} - B_k^\ell U^\ell_k \|^2 \right]
\]

\[
= \text{tr} \left( A_\ell E_{X^\ell,X^\ell} A_\ell^T - 2A_\ell E_{X^\ell,X^\ell} + E_{X^\ell,X^\ell} \right)
\]

\[
+ \sum_{k=1}^{K} \left( (B_k^\ell)^T \mathbb{E} \left[ (U_k^\ell)^2 \right] B_k^\ell \right) - 2(B_k^\ell)^T \left( \mathbb{E} \left[ U_k^\ell X^\ell_k \right] - A_\ell \mathbb{E} \left[ U_k^\ell X^\ell_{k-1} \right] \right).
\]

(12.39)

Thus, we have

\[
\hat{B}_k^\ell = \text{argmin}_{B_k^\ell \in \mathcal{B}_\ell} (B_k^\ell)^T \mathbb{E} \left[ (U_k^\ell)^2 \right] B_k^\ell - 2(B_k^\ell)^T \left( \mathbb{E} \left[ U_k^\ell X^\ell_k \right] - A_\ell \mathbb{E} \left[ U_k^\ell X^\ell_{k-1} \right] \right).
\]

(12.40)

Assuming an unconstrained \( B_k^\ell \) (i.e., \( \mathcal{B}_\ell = \mathbb{R}^{n_\ell} \)), we have

\[
\hat{B}_k^\ell = \frac{\mathbb{E} \left[ U_k^\ell X^\ell_k \right] - \hat{A}_\ell \mathbb{E} \left[ U_k^\ell X^\ell_{k-1} \right]}{\mathbb{E} \left[ (U_k^\ell)^2 \right]},
\]

(12.41)

for \( k \in \mathcal{K}_\ell \), with

\[
\mathcal{K}_\ell = \left\{ k \in \{1, \ldots, K\} : \mathbb{E} \left[ (U_k^\ell)^2 \right] \neq 0 \right\}.
\]

(12.42)

Note that when \( \mathbb{E} \left[ (U_k^\ell)^2 \right] = 0 \), it implies that \( \hat{\Sigma}_{U_k^\ell} = 0 \) in the previous EM step, and thus \( \mathbb{E} [U_k^\ell X_k^\ell] = \mathbb{E} [U_k^\ell X_{k-1}^\ell] = 0 \). Consequently, for such \( k \in \mathcal{K}_\ell \), \( \hat{B}_k^\ell \) is undefined since it has no effect anymore on the likelihood function.
The parameter $\hat{A}_\ell$ appearing in (12.41) is found as in (12.30) with

$$W_\ell = \mathcal{E}_{X^\ell, X} - \sum_{k \in \mathcal{K}_\ell} \frac{\mathbb{E}[U_k^\ell X_{k-1}^\ell] \mathbb{E}[U_k^\ell (X_{k-1}^\ell)^T]}{\mathbb{E}[(U_k^\ell)^2]} \quad (12.43)$$

$$\xi_\ell = \mathcal{E}_{X^\ell, X} - \sum_{k \in \mathcal{K}_\ell} \frac{\mathbb{E}[U_k^\ell X_{k-1}^\ell] \mathbb{E}[U_k^\ell (X_k^\ell)^T]}{\mathbb{E}[(U_k^\ell)^2]}, \quad (12.44)$$

which we obtain by plugging (12.41) in (12.39).

Constraints on $B_k^\ell$ can be included in (12.40) but at the price of potentially giving up the closed-form expression for $A_\ell$. In that case, it is however possible to alternate EM updates where either $A_\ell$ or $B_k^\ell$ (but not both) is updated at a time.

### 12.2.3 Filtered White Gaussian Noise

Following Sec. 10.3.5, the parameters are such that $B_k^\ell = 0$ for all $k$.

The maximization of (12.25) reduces to minimizing

$$\text{tr} \left( (A_\ell \mathcal{E}_{X^\ell, X} A_\ell^T - A_\ell \mathcal{E}_{X^\ell, X} - (A_\ell \mathcal{E}_{X^\ell, X}^T + \mathcal{E}_{X, X}) V_{E^\ell}^{-1} \right)$$

$$+ K \ln |V_{E^\ell}|, \quad (12.45)$$

where we used a similar expansion as in (12.27).

Cases of interest are actually when $A_\ell$ is known, which we thus assume. Then, $\hat{V}_{E^\ell}$ which minimizes (12.45) is given by

$$\hat{V}_{E^\ell} = \frac{1}{K} \left( A_\ell \mathcal{E}_{X^\ell, X} A_\ell^T - A_\ell \mathcal{E}_{X^\ell, X} - (A_\ell \mathcal{E}_{X^\ell, X}^T + \mathcal{E}_{X, X}) \right). \quad (12.46)$$

Other examples of interest are when $V_{E^\ell}$ has a specific structure. For instance, for spline smoothing, the covariance matrix writes as

$$V_{E^\ell} = \sigma_{E^\ell}^2 V_0, \quad (12.47)$$

for some known matrix $V_0 \in \mathbb{S}_+^n$ and unknown smoothing variance $\sigma_{E^\ell}^2$. It follows that

$$\hat{\sigma}_{E^\ell}^2 = \frac{1}{K n_\ell} \text{tr} \left( V_0^{-1} (A_\ell \mathcal{E}_{X^\ell, X} A_\ell^T - 2A_\ell \mathcal{E}_{X^\ell, X} + \mathcal{E}_{X, X}) \right). \quad (12.48)$$

It is also straightforward to restrict the state noise covariance matrix to be diagonal, i.e.,

$$V_{E^\ell} = \text{diag} \left( \{ \sigma_{E^\ell}^2 \}_{1, \ldots, n_\ell} \right), \quad (12.49)$$
leading to the EM update, for \( q \in \{1, \ldots, n_\ell \} \),
\[
\{ \hat{\sigma}^2_{E[i]} \}_q = \frac{1}{K} \left\{ A_\ell \mathcal{E}_{X^\ell, X} - 2A_\ell \mathcal{E}_{X^\ell, X} + \mathcal{E}_{X, X} \right\}_q.
\] (12.50)

### 12.3 Alternative EM Algorithms for Restrictive Cases

The EM algorithm described in Sec. 12.2 handles the estimation of all input and system parameters with closed-form EM updates (unless unwieldy constraints are introduced). However, two remarks are worth mentioning. First, the EM algorithm described in Sec. 12.2 does not create exact zeros for the input variances. Secondly, the choice of hidden variables for the EM algorithm is not unique. It is difficult not to pick \( X = (X_1, \ldots, X_K) \) as hidden variables since otherwise the EM update for \( A \) and \( B_k, k \in \{1, \ldots, K\} \) would become impractical. Nonetheless, instead of using both \( X = (X_1, \ldots, X_K) \) and \( U = (U_1, \ldots, U_K) \) as hidden variables, we could only choose a subset of all those variables and/or use other hidden variables.

#### 12.3.1 Using State Variables Only

For a given signal component \( \ell \in \{1, \ldots, L\} \), we might want to use \( X^\ell = (X_1^\ell, \ldots, X_K^\ell) \) only as hidden variable and integrate out \( U^\ell = (U_1^\ell, \ldots, U_K^\ell) \). By doing such a modification for a specific component \( \ell \in \{1, \ldots, L\} \), the new EM update consists of finding
\[
\left\{ A_\ell, B_1^\ell, \ldots, B_K^\ell, V_{E[\ell]}, \Sigma_{U_1^\ell}, \ldots, \Sigma_{U_K^\ell} \right\},
\]
which maximize
\[
\sum_{k=1}^{K} \mathbb{E} \left[ \ln \mathcal{N} \left( X^\ell_k : A_\ell X^\ell_{k-1}, V_{E[\ell]} + B_k^\ell \Sigma_{U_k^\ell} (B_k^\ell)^T \right) \right].
\] (12.51)

This maximization has no general closed-form solution and is equivalent to minimizing
\[
J(\theta_\ell) = \sum_{k=1}^{K} \left( \text{tr} \left( \left( V_{E[\ell]} + B_k^\ell \Sigma_{U_k^\ell} (B_k^\ell)^T \right)^{-1} V_{E[\ell]} \right) \right.
+ \ln \left| V_{E[\ell]} + B_k^\ell \Sigma_{U_k^\ell} (B_k^\ell)^T \right|.
\] (12.52)
with

\[ V_k^{[\ell]} = \mathbb{E} \left[ (X_k^{[\ell]} - A_\ell X_{k-1}^{[\ell]}) (X_k^{[\ell]} - A_\ell X_{k-1}^{[\ell]})^T \right]. \tag{12.53} \]

Nonetheless, in the special case where \( A_\ell \) is fixed, \( V_{E^{[\ell]}} = \sigma_{E^{[\ell]}}^2 I_{n_\ell} \), \( j_\ell = 1 \) (and thus \( \Sigma_{U_k^{[\ell]}} = \sigma_{U_k^{[\ell]}}^2 \in \mathbb{R}_+ \)), and \( B_k^{[\ell]} \in \mathbb{R}^{n_\ell} \) (i.e., \( B_\ell = \mathbb{R}^{n_\ell} \)), we have

\[
J(\theta_\ell) = \sum_{k=1}^{K} \left( \frac{\text{tr}(V_k^{[\ell]})}{\sigma_{E^{[\ell]}}^2} + n_\ell \ln(2\pi\sigma_{E^{[\ell]}}^2) \right) - \frac{\sigma_{U_k^{[\ell]}}^2 (B_k^{[\ell]})^T V_k^{[\ell]} B_k^{[\ell]}}{\sigma_{E^{[\ell]}}^2 \left( \sigma_{E^{[\ell]}}^2 + \sigma_{U_k^{[\ell]}}^2 \right)} + \ln \left( \frac{\sigma_{E^{[\ell]}}^2 + \sigma_{U_k^{[\ell]}}^2}{\sigma_{E^{[\ell]}}^2} \right), \tag{12.54}
\]

using (A.4) and (A.5), and where we have assumed that \( \|B_k^{[\ell]}\| = 1 \) since there is an undetermined scale factor between \( \sigma_{U_k^{[\ell]}} \) and \( B_k^{[\ell]} \). By maximizing this expression, we obtain

\[
\hat{B}_k^{[\ell]} = \arg\max_{\|B_k^{[\ell]}\|=1} (B_k^{[\ell]})^T V_k^{[\ell]} B_k^{[\ell]} . \tag{12.55}
\]

Thus, \( \hat{B}_k \) is the unit eigenvector corresponding to the maximum eigenvalue \( \lambda_k^{[\ell]} \) of \( V_k^{[\ell]} \). Then, we also have

\[
\hat{\sigma}_{U_k^{[\ell]}}^2 = \max \left( 0, \lambda_k^{[\ell]} - \sigma_{E^{[\ell]}}^2 \right) \tag{12.56}
\]

\[
\hat{\sigma}_{E^{[\ell]}}^2 = \arg\min_{\sigma_{E^{[\ell]}}^2} \left( \sum_{k=1}^{K} \frac{\text{tr}(V_k^{[\ell]})}{\sigma_{E^{[\ell]}}^2} + n_\ell K \ln(\sigma_{E^{[\ell]}}^2) \right) + \sum_{k: \lambda_k^{[\ell]} > \sigma_{E^{[\ell]}}^2} - \frac{\lambda_k^{[\ell]}}{\sigma_{E^{[\ell]}}^2} + 1 + \ln \left( \frac{\lambda_k^{[\ell]}}{\sigma_{E^{[\ell]}}^2} \right) . \tag{12.57}
\]

This last optimization problem can be solved by considering the \( K + 1 \) sub-problems of restricting \( \sigma_{E^{[\ell]}}^2 \) in the interval of consecutive eigenvalues (among the \( \lambda_k^{[\ell]}, k \in \{1, \ldots, K\} \)) ranked in decreasing order and selecting the optimum. To avoid such complications, we can also use an approximate step by replacing \( \sigma_{E^{[\ell]}}^2 \) in the sum index by the previous estimate \( (\sigma_{E^{[\ell]}}^2)_{\text{old}} \).

The charm of this alternative EM update is that exact zeros can occur due to the update (12.56). Nevertheless, this alternative EM algorithm is only applicable for a very restrictive case.
12.3.2 Using an Alternative Input Variable

For a signal component $\ell \in \{1, \ldots, L\}$, we here use $X^{[\ell]} = (X_1^{[\ell]}, \ldots, X_K^{[\ell]})$ and $T^{[\ell]} = (T_1^{[\ell]}, \ldots, T_K^{[\ell]})$ as hidden variables. The hidden variables $T_k^{[\ell]}$ are defined such that

$$p(u_k^{[\ell]} | \Sigma_{U_k^{[\ell]}}) = \mathcal{N}(u_k^{[\ell]} : 0, \Sigma_{U_k^{[\ell]}})$$

$$= \int \delta(u_k^{[\ell]} - S_{U_k^{[\ell]}} t_k^{[\ell]}) \mathcal{N}(t_k^{[\ell]} : 0, I_{j\ell}) \, dt_k^{[\ell]},$$

for all $k \in \{1, \ldots, K\}$ and where

$$S_{U_k^{[\ell]}} = \text{diag}(\sigma_{U_k^{[\ell]}}).$$

By doing such a modification for a specific component $\ell \in \{1, \ldots, L\}$, the new EM update consists of finding

$$\{A_\ell, B_1^{[\ell]}, \ldots, B_K^{[\ell]}, V_{E^{[\ell]}}, S_{U_1^{[\ell]}}, \ldots, S_{U_K^{[\ell]}}\},$$

which maximizes

$$\sum_{k=1}^K \mathbb{E} \left[ \ln \mathcal{N} \left( X_k^{[\ell]} : A_\ell X_{k-1}^{[\ell]} + B_k^{[\ell]} S_{U_k^{[\ell]}} T_{U_k^{[\ell]}}, V_{E^{[\ell]}} \right) \right].$$

This maximization problem is similar to the maximization of (12.25) with the substitution $B_k^{[\ell]} \leftarrow B_k^{[\ell]} S_{U_k^{[\ell]}}$. However, the constraints introduced on the $B_k^{[\ell]}$ for $k \in \{1, \ldots, K\}$ are in general much harder to handle. For instance, when $B_k^{[\ell]}$ is independent of $k$, this alternative EM update becomes impractical. Nevertheless, in some restrictive cases such as $A_\ell$ and $B_k^{[\ell]}$ all fixed, the EM update involving the maximization of (12.61) might be of great use since (12.61) is a quadratic form in $S_{U_k^{[\ell]}}$.

12.4 Marginal Likelihood Update

We here study the marginal likelihood with respect to a single input parameter $\sigma_{U_k^{(j)}}^2$ for some $k \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, J\}$. This marginal likelihood is defined as

$$p(y; \sigma_{U_k^{(j)}}^2) \propto p(y | \theta, \sigma_U^2),$$

(12.62)
Figure 12.2 – Equivalent factor graph representation of Fig. 12.1 with inputs split element-wise. \( U^{(j)}_k \) is the \( j^{th} \) element of \( U_k \) and \( b_{k,j} \) denotes the \( j^{th} \) column of \( B_k \).

where \( \propto \) means equality up to a scale factor independent of \( \sigma^2_{U_k^{(j)}} \) and where \( p(y|\theta, \sigma^2_U) \) is the likelihood function (12.8). Note that \( \theta \) and the remaining parameters in \( \sigma^2_U \) are assumed to be fixed to some value. Moreover, while each component of \( \sigma^2_U \) can be individually updated by maximizing its marginal likelihood, there is in general no trivial marginal update for the elements of \( \theta \).

A helpful factor graph representation that is equivalent to Fig. 12.1 is given in Fig. 12.2. The difference is that the input \( U_k \) is further split into its components \( U_k^{(j)}, j \in \{1, \ldots, J\} \).

12.4.1 Mean/Variance Parameterization

Using the notion of closing boxes [45] in the factor graph of Fig. 12.2 which represents the joint density \( p(y, x, u|\theta, \sigma^2_U) \), the marginal likelihood readily writes as

\[
p(y; \sigma^2_{U_k^{(j)}}) \propto p(y|\theta, \sigma^2_U)
\]

\[
\propto \int \mathcal{N}(u_k^{(j)}: \overrightarrow{m}_{U_k^{(j)}}, \overrightarrow{V}_{U_k^{(j)}}) \mathcal{N}(u_k^{(j)}: 0, \sigma^2_{U_k^{(j)}}) \, du_k^{(j)} \quad (12.64)
\]
\[ \propto \mathcal{N}(\hat{m}_{U_k^{(j)}} : 0, \hat{V}_{U_k^{(j)}} + \sigma^2_{U_k^{(j)}}), \]  

where \( \hat{m}_{U_k^{(j)}} \) and \( \hat{V}_{U_k^{(j)}} \) are the mean and variance of the upward Gaussian message on edge \( U_k^{(j)} \). Thus, maximizing the marginal likelihood function leads to

\[ \hat{\sigma}^2_{U_k^{(j)}} = \max \left( 0, \hat{m}^2_{U_k^{(j)}} - \hat{V}_{U_k^{(j)}} \right). \]  

This update step can create exact zeros. Thus, in order to decide whether \( \hat{\sigma}^2_{U_k^{(j)}} \) is an exact zero or not once the EM algorithm has almost converged, a marginal likelihood update can be used as a smart thresholding function.

### 12.4.2 Mean/Precision Parameterization

An interesting alternative parameterization of (12.66) is to use the variable \( \tilde{U}_k^{(j)} \) (cf., Fig. 12.2). Using the backward mean and precision value \( [45] \) on edge \( U_k^{(j)} \) and denoting \( b_{k,j} \) the \( j^{th} \) column of \( B_k \), we have

\[ \hat{V}_{U_k^{(j)}} = (b_{k,j}^T \tilde{W}_{U_k^{(j)}} b_{k,j})^{-1}, \]  

and

\[ \hat{m}_{U_k^{(j)}} = \tilde{V}_{U_k^{(j)}} \tilde{W}_{U_k^{(j)}} \hat{m}_{U_k^{(j)}} \]

\[ = (b_{k,j}^T \tilde{W}_{U_k^{(j)}} b_{k,j})^{-1} b_{k,j}^T \tilde{W}_{U_k^{(j)}} \hat{m}_{U_k^{(j)}}. \]

Noticing that

\[ \hat{m}_{U_k^{(j)}} = -\tilde{m}_{X_k} + \hat{m}_{X_k} \]

\[ = \mu_{X_k} \]

is independent of index \( j \) (since all inputs are zero-mean), (12.66) also writes as

\[ \hat{\sigma}^2_{U_k^{(j)}} = \max \left( 0, \frac{\left( b_{k,j}^T \tilde{W}_{U_k^{(j)}} \mu_{X_k} \right)^2 - b_{k,j}^T \tilde{W}_{U_k^{(j)}} b_{k,j}}{(b_{k,j}^T \tilde{W}_{U_k^{(j)}} b_{k,j})^2} \right). \]
Thus, an input is introduced (i.e., $\hat{\sigma}^2_{U_k^{(j)}} \neq 0$) if and only if

$$
(b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} \mu_{X_k})^2 > b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} b_{k,j},
$$

(12.73)

that is to say, if and only if, the discrepancy $\mu_{X_k}$ between the forward and backward messages projected into the direction $b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}}$ is big enough. Consequently, a non-zero variance is introduced only to compensate a substantial mismatch between the forward (using observations up to index $k$ only) and backward (using observations from index $k+1$ and so forth) estimates.

Using Cauchy-Schwarz inequality, we have

$$
(b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} \mu_{X_k})^2 \leq (b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} b_{k,j}) (\mu_{X_k}^T \hat{\mathbf{W}}_{U_k^{(j)}} \mu_{X_k}).
$$

(12.74)

As a consequence, a necessary but clearly not sufficient condition for introducing a non-zero input is

$$
\mu_{X_k}^T \hat{\mathbf{W}}_{U_k^{(j)}} \mu_{X_k} > 1.
$$

(12.75)

Note that in terms of the backward mean and precision value, the posterior mean $m_{U_k^{(j)}}$ of $U_k^{(j)}$ is obtained with (4.40), which also expresses as

$$
m_{U_k^{(j)}} = -\frac{\sigma^2_{U_k^{(j)}} b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} \mu_{X_k}}{1 + \sigma^2_{U_k^{(j)}} b^T_{k,j} \hat{\mathbf{W}}_{U_k^{(j)}} b_{k,j}},
$$

(12.76)

using similar expansions as above.

### 12.4.3 Dual Mean/Dual Precision Parameterization

We now express (12.66) in terms of the parameterization of the modified Bryson-Frazier smoother. Indeed, using the message passing rules in \[44\] and denoting $b_{k,j}$ the $j^{th}$ column of $B_k$, we obtain

$$
\hat{m}_{U_k^{(j)}} = -\hat{W}_{U_k^{(j)}}^{-1} \hat{\xi}_{U_k^{(j)}}
$$

(12.77)

$$
= -(b^T_{k,j} \hat{\mathbf{W}}_{X_k} b_{k,j})^{-1} b^T_{k,j} \hat{\xi}_{X_k},
$$

(12.78)

and

$$
\hat{V}_{U_k^{(j)}} = \hat{W}_{U_k^{(j)}}^{-1} - \sigma^2_{U_k^{(j)}}
$$

(12.79)

$$
= (b^T_{k,j} \hat{\mathbf{W}}_{X_k} b_{k,j})^{-1} - \sigma^2_{U_k^{(j)}}.
$$

(12.80)
Even if both quantities (12.78) and (12.80) seem to depend on $\sigma^2_{U_k(j)}$ (either directly or via $\tilde{\xi}_X$ and $\tilde{W}_X$), they actually do not. This appearing dependency is only related to the message parameterization. Then, (12.66) writes as

$$
\hat{\sigma}^2_{U_k(j)} = \max \left( 0, \frac{(b_{k,j}^{T}\tilde{\xi}_X)^2 - b_{k,j}^{T}\tilde{W}_X b_{k,j}}{(b_{k,j}^{T}\tilde{W}_X b_{k,j})^2} + \sigma^2_{U_k(j)} \right),
$$

(12.81)

for any value of $\sigma^2_{U_k(j)}$ which is also indirectly used for the computation of $\tilde{\xi}_X$ and $\tilde{W}_X$.

In particular, at the optimum value $\sigma^2_{U_k(j)} = \hat{\sigma}^2_{U_k(j)}$, (12.81) must hold.

If $\hat{\sigma}^2_{U_k(j)} = 0$, we conclude that

$$
(b_{k,j}^{T}\tilde{\xi}_X)^2 \leq b_{k,j}^{T}\tilde{W}_X b_{k,j},
$$

(12.82)

when $\sigma^2_{U_k(j)} = \hat{\sigma}^2_{U_k(j)}$. If $\hat{\sigma}^2_{U_k(j)} \neq 0$, we deduce that

$$
(b_{k,j}^{T}\tilde{\xi}_X)^2 = b_{k,j}^{T}\tilde{W}_X b_{k,j},
$$

(12.83)

when $\sigma^2_{U_k(j)} = \hat{\sigma}^2_{U_k(j)}$. 

Chapter 13

Relevant Applications of Learning Sparse Signal Decompositions

13.1 Learning Repetitive Signal Shapes

Many signals and in particular raw physiological signals are composed of a superposition of a wandering baseline along with several repetitive, possibly overlapping signal shapes produced by different sources. For instance, an abdominal electrocardiogram (ECG) of a pregnant woman, as in the upper plot of Fig. 13.4, is composed of the maternal ECG, the fetal ECG, and a wandering baseline mainly caused by the mother’s breathing. The goal is to decompose a single-channel signal into a baseline and unknown but repetitive signal shapes. This includes learning the signal shapes and finding their occurrences. The proposed method outputs a sparse multi-channel representation of the given signal, which can be interpreted as a signal labeling.

In this section, we exclusively focus on single-channel observations (i.e., $M = 1$) even if the multi-channel extension is straightforward. The problem description and some of the examples are part of [87].
13.1.1 Problem Setup

Let \( y = (y_1, \ldots, y_K) \in \mathbb{R}^K \) be a given single-channel discrete-time signal of length \( K \in \mathbb{N} \). We wish to explain the signal \( y \) as a superposition of a smooth baseline \( s_k^{[0]} \in \mathbb{R} \) and \( L \in \mathbb{N} \) signal components \( s_k^{[\ell]} \in \mathbb{R} \), \( \ell \in \{1, \ldots, L\} \), \( k \in \{1, \ldots, K\} \), with each component consisting of repetitions (across time) of an unknown or known signal shape. Occurrences of a signal shape can vary in amplitude and overlap with other signal shapes, including its own. Specifically, we have

\[
y_k = s_k^{[0]} + \sum_{\ell=1}^{L} s_k^{[\ell]} + Z_k, \quad (13.1)
\]

for \( k \in \{1, \ldots, K\} \), with observation noise \( Z_k \sim \mathcal{N}(0, \sigma_Z^2) \). Note that (13.1) actually corresponds to (10.1) where the \( D \) matrix is simply a row vector consisting of ones.

Each signal component \( s^{[\ell]} \) is modeled with its own and unknown LSSM as in (10.2). For the signal components corresponding to repetitive signal shapes (i.e., \( \ell > 0 \)), we use the single signal shape model as described in Sec. 10.3.1 with \( j_\ell = q_\ell = 1 \). This model assumption simply means that \( s^{[\ell]} \) is composed of few scaled and time-shifted versions of a single shape, up to additive state noise terms.

For the smooth baseline model (i.e., \( \ell = 0 \)), we use a filtered white Gaussian noise model as described in Sec. 10.3.5. Surprisingly, the baseline model is the sparsest signal component in the representation (10.2). For that reason, it is rarely included within the BSS problem but rather filtered out in a pre-processing step.

Finally, combining the LSSMs of the signal components, the signal \( y \) is the output of a joint LSSM of order \( n = \sum_{\ell=0}^{L} n_\ell \) and as in (10.5).

Thus, decomposing a single-channel signal into a baseline and repetitive unknown signal shapes reduces to the estimation of the input and system parameters of the joint model. Those parameters are estimated by maximum likelihood using an efficient EM algorithm, as described in Chapter 12.

13.1.2 Synthetic Example of Signal Separation and Learning

We first illustrate our algorithm with a synthetic example. We generate a signal as in the upper plot of Fig. 13.1 which superimposes a baseline
generated with filtered white Gaussian noise, two different decaying sinusoids, spikes, an offset, and white Gaussian noise. For our algorithm, we use a LSSM with $L = 4$ where

- $n_0 = 2$ with $(C_0, A_0, V_0)$ that models a cubic spline smoothing [61],
- $n_1 = 2$ with $C_1 = [1 \ 0]$ and unknown $A_1 = \rho_1 R(\omega_1)$ and $B_1 \in \mathbb{R}^2$ to model a decaying sinusoid,
- $n_2 = 2$ with $C_2 = [1 \ 0]$ and unknown $A_2 = \rho_2 R(\omega_2)$ and $B_2 \in \mathbb{R}^2$ to model another decaying sinusoid,
- $n_3 = 1$ with $C_3 = B_3 = 1$ and $A_3 = 0$ to model spikes,
- $n_4 = 1$ with $C_4 = A_4 = B_4 = 1$ to model offsets.

While $A_1$, $B_1$, $A_2$, and $B_2$ are randomly initialized, the input variances are initialized as $\sigma_{E[\ell]}^2 = 10^{-5}$, $\sigma_{U_k[1]}^2 = \sigma_{U_k[2]}^2 = 10^{-6}$, and $\sigma_{U_k[3]}^2 = \sigma_{U_k[4]}^2 = 10^{-7}$ in order to favor the unknown models over the known (spike and offset) ones. The noise variance $\sigma_Z^2$ is fixed to $10^{-1}$. 

Figure 13.1 – Synthetic example of learning a sparse signal decomposition.
As can be seen in Fig. 13.1, our algorithm recovers the individual input positions of each model (indicated in the second plot by the non-zero values of $\hat{\sigma}_{U_k}$) and outputs a good estimation for both the individual signal shape components $\hat{s}^{[1]}$ and $\hat{s}^{[2]}$ and the baseline $\hat{s}^{[0]}$. If more unknown shapes (i.e., $L > 4$) than actually present are specified, unnecessary signal shapes are automatically disregarded by the algorithm under adequate initialization of parameters.

13.1.3 ECG Signal Separation and Learning

In this example, we want to decompose an ECG recording (e.g., lead II of a standard recording as in Fig. 13.2, upper plot) into a wandering baseline and repetitive ECG signal shapes. The wandering baseline is mainly caused by the patient’s breathing while the ECG signal shape is the result of the patient’s heart electrical cycle.

For our algorithm, we use a single signal component (i.e., $L = 1$)
13.1 Learning Repetitive Signal Shapes

consisting of only two decaying sinusoids and a baseline component consisting of a cubic spline model. The initial values are set to $\sigma_Z^2 = 8 \cdot 10^{-2}$, $\sigma_{U_1}^2 = 10^{-4}$, $\sigma_{E_0}^2 = 10^{-5}$, and $\sigma_{E_1}^2 = 10$. In Fig. 13.2, we display the results of our algorithm on a standard ECG recording from the DaISy dataset [23]. All occurrences of the ECG signal shape are correctly detected. Furthermore, both the baseline component (blue line in the upper plot) and the signal component corresponding to repetitions of the ECG shape (green line in the lower plot) are nicely separated and estimated.

In Fig. 13.3, we zoom in on one ECG signal shape (around time index $k = 1100$). This plot confirms that the estimated ECG shape is suitably estimated with our algorithm using only two decaying sinusoids.

13.1.4 Fetal ECG Signal Separation and Learning

In Fig. 13.4, we display the result of our algorithm on a raw abdominal ECG recording of a pregnant woman from the DaISy dataset [23]. In this recording, the fetal ECG signal is about eight times weaker than the maternal ECG, which thus consists of a delicate signal separation and estimation problem. For our algorithm, we use $L = 2$ models consisting of linear combinations of 8 and 3 damped sinusoids. The baseline model still emulates cubic spline smoothing. The weak fetal ECG signal (plotted in purple) is remarkably well estimated and separated from the strong maternal ECG signal (plotted in green), as indicated in the subplots 2 & 3 of Fig. 13.4.

We also compare our algorithm with an adaptation of the K-SVD algorithm [2] that finds $L$ sparse vectors $w^{(\ell)} \in \mathbb{R}^K$ and dictionaries
Figure 13.4 – Fetal ECG signal separation and learning. 1) Raw ECG. 2,3) Estimated signal components from our algorithm. 4,5) Estimated signal components from $K$-SVD.
\[ y_k \hat{s}[0]_k \hat{s}[1]_k 1,000 1,100 1,200 \]

\[ y_k \hat{w}(1) \hat{w}(1) \]

\[ H_\ell \in \mathbb{R}^{K \times K} \] consisting of time-shifted versions of a single vector, while minimizing \( \| y - \sum_{\ell=1}^{L} H_\ell w^{(\ell)} \| \) subject to, \( w^{(\ell)} \geq 0, \| w^{(\ell)} \|_1 \leq T_\ell \), for some fixed \( T_\ell > 0 \). The results, for \( L = 2 \), are shown in subplots 4 & 5 of Fig. 13.4. While the maternal ECG signal is well estimated, the fetal ECG signal is poorly separated and estimated. Indeed, the \( K \)-SVD algorithm does not allow much variation between occurrences of a signal shape. Thus, despite a good estimation of the maternal ECG signal, the remaining errors are still large enough to spoil the estimation and separation of the weak fetal ECG (plotted in purple in subplots 4 & 5 of Fig. 13.4). In this regard, our method is more robust since small variations of a pulse shape are compensated by state noise terms (via the variances \( \sigma^2_{E[\ell]} \)).

We also apply both methods on an ECG recording where the fetal

**Figure 13.5** – Estimated maternal ECG signal (green line) using the proposed algorithm (upper plot) and \( K \)-SVD (lower plot).
ECG signal is about two times weaker than the maternal ECG signal. In this scenario, both algorithms successfully estimate and separate the two signal components (not reported here). However, as shown in Fig. 13.5, the maternal ECG signal is estimated in different ways for each method. Namely, our algorithm includes the P and T waves of the maternal ECG in the baseline component, while $K$-SVD algorithm neglects the T waves and merges the P waves with the QRS complex. Since the time delays between the P waves, QRS complexes, and T waves vary according to the heart rate (which also has some variability), it is actually worthwhile to include the T and P waves in the baseline component, which in turn allows a more robust estimation of the other signals.

### 13.2 Learning Classes of Signal Shapes

We want to decompose a multi-channel signal into several signal components, where each signal component consists of occurrences of signal shapes from some class. If the class of signal shapes reduces to a single signal shape, we obtain the “repetitive signal shape” model described in Sec. 13.1.

#### 13.2.1 Problem Setup

Consider $L \in \mathbb{N}$ classes of signal shapes as described in Sec. 10.3.3. Each class is associated with its given set of constraints $B_\ell$ and its signal component $s_k^{[\ell]} \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$, to be estimated.

We wish to decompose a multi-channel signal $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$ using signal shapes from these $L$ classes. Specifically, we have

$$y_k = \sum_{\ell=1}^L s_k^{[\ell]} + Z_k,$$  

(13.2)

for $k \in \{1, \ldots, K\}$, with observation noise $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2 I_M)$. Note that (13.2) actually corresponds to (10.1) where the matrix $D \in \mathbb{R}^{M \times (ML)}$ consists of stacking (horizontally) $L$ identity matrices $I_M$.

The goal is to estimate the $L$ classes of signal shapes. In other words, for each class, we want to estimate its LSSM parameters and find the occurrences of its signal shapes.

Note that a baseline signal component can easily be added to our model.
13.2 Learning Classes of Signal Shapes

13.2.2 System Learning

Consider a signal generator which can be triggered in different and infinitely many ways. Given a recording of the response of that system to several successive triggers, we wish to estimate the system parameters. This is a general blind system identification problem where both the input occurrences and the input directions are unknown [85].

To solve this problem, we consider only one class of signal shapes (i.e., \( L = 1 \)) which we model with a LSSM as in Sec. 10.3.3 of order \( n_1 \in \mathbb{N} \) with unconstrained input vectors (i.e., \( \mathcal{B}_1 = \mathbb{R}^{n_1} \)). Thus, the characteristics of the signal generator are encoded into the poles of the unknown matrix \( A_1 \). Since the number of times that the system is triggered is small compared to the number of signal samples, the sparsity assumption implied by our model is justified.

In Fig. 13.6, we show the result of our algorithm on a simulated example. We use a system which can generate three types of decaying sinusoids (three poles) with discrete angular frequencies and decays

\[(\omega, \rho) \in \{(0.1047, 0.990), (0.1257, 0.992), (0.2094, 0.991)\}. \quad (13.3)\]

The system is triggered five times (indicated by gray circles whose \( y \)-coordinate is the input amplitude) and generates the noisy signal plotted

\[ y_k \]

\[ \hat{\sigma}^{[1]}_{U_k} \]
For our algorithm, we use a LSSM which consists of three decaying sinusoids \( n_1 = 6 \). While the frequencies and decays are randomly initialized, the other parameters are initialized to \( \sigma^2_{U_k^{[1]}} = 10^{-12} \) and \( \sigma^2_{E^{[1]}} = 1 \). The variance \( \sigma_Z^2 \) is fixed to \( 6 \cdot 10^{-3} \).

We observe that the estimated signal component \( \hat{s}^{[1]}_k \) (red line) accurately captures the dynamic of the system even if the inputs are not perfectly matched to the actual ones (red bars, lower plot). In particular, the estimated poles of the system

\[
(\hat{\omega}, \hat{\rho}) \in \{(0.1063, 0.993), (0.1228, 0.993), (0.2086, 0.990)\},
\]

\( (13.4) \)

are very close to the actual ones. However, the second and third actual inputs are estimated with two or three smaller amplitude inputs instead of a single one with higher amplitude. For this blind system identification problem, this behavior is quite typical and is delicate to be avoided since the direction of the input changes at each triggering point. In addition, a greater value of \( \sigma_Z^2 \) would increase the sparsity level and thus split inputs would tend to disappear. Nevertheless, since we also learn the system, a too high value of \( \sigma_Z^2 \) might also drift the estimated system to an undesirable one.

Note also that our algorithm converges to a local optimum that might not be good enough according to our expectations. To deal with this issue, several restarts of the algorithm could be performed and the best result can be selected.

### 13.2.3 Multi-Sinusoidal System

We simulate a noisy signal composed of a superposition of a wandering baseline (filtered white Gaussian noise) and several occurrences of three decaying sinusoids of decays and angular frequencies

\[
\hat{\rho}_1 = 0.988 , \quad \hat{\omega}_1 = 0.12566371 \\
\hat{\rho}_2 = 0.989 , \quad \hat{\omega}_2 = 0.20943951 \\
\hat{\rho}_3 = 0.995 , \quad \hat{\omega}_3 = 0.10471976.
\]

\( (13.5) \) \( (13.6) \) \( (13.7) \)

Each time a decaying sinusoidal shape appears, it comes with its own amplitude and phase.

Given such signals as in Fig. \( 13.7 \) upper plot, we want to learn a signal decomposition. Namely, we wish to estimate the baseline, learn
Figure 13.7 – Signal decomposition of three sinusoidal systems triggered several times, each time with a different phase and amplitude.

the frequencies and decays of each sinusoidal component and find the occurrences of each sinusoidal signal shape.

This signal decomposition and learning problem fits to the one described in Sec. 13.2.1. Thus, we use $L = 3$ signal components consisting of few signal shapes from a signal class. Each signal class is specified to
be all decaying sinusoids of some frequency and decay with any amplitude and phase. However, neither the parameters of the sinusoid (i.e., frequency and decay) nor the starting time, amplitude, and phase of each occurrence are known. Thus, all parameters need to be estimated. More precisely, the LSSM parameters of the $\ell$th signal component $s^{[\ell]}$, $\ell \in \{1, 2, 3\}$, are

\[
\begin{align*}
    n_\ell &= 2 \\
    C_\ell &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\
    A_\ell &= \rho_\ell R(\omega_\ell) \\
    B_\ell &= \mathbb{R}^2,
\end{align*}
\]

where $\omega_\ell$ and $\rho_\ell$ need to be estimated. We also add a baseline signal component $s^{[0]}$ which emulates a cubic spline smoothing with unknown smoothing variance.

For our algorithm, we set $\sigma^2_Z = 2 \cdot 10^{-3}$ and initialize $\sigma^2_{U[k]} = 10^{-12}$ and $\sigma^2_{E[\ell]} = 0.1$, for $\ell \in \{1, 2, 3\}$. In Fig. 13.7 we display the result of our algorithm. The three sinusoidal components $\hat{s}^{[\ell]}$, $\ell \in \{1, 2, 3\}$ and the baseline signal $\hat{s}^{[0]}$ are flawlessly separated and estimated, as reflected by the decay and frequency estimates

\[
\begin{align*}
    \tilde{\rho}_1 &= 0.987977, \quad \tilde{\omega}_1 = 0.125598 \\
    \tilde{\rho}_2 &= 0.988903, \quad \tilde{\omega}_2 = 0.209369 \\
    \tilde{\rho}_3 &= 0.99492, \quad \tilde{\omega}_3 = 0.104858,
\end{align*}
\]

which are really close to the actual ones. Moreover, all signal shape occurrences are detected and assigned to its correct class, as shown by the non-zero values of $\hat{\sigma}_{U[k]}$ in the second plot of Fig. 13.7. The actual signal decomposition used to generate the measurements fit the estimated one almost perfectly and would be indistinguishable if plotted in Fig. 13.7.

### 13.2.4 Model Switches

In this part, we provide an algorithm to detect model switches in discrete-time observations. An alternative approach was already proposed in Sec. 8.6.

#### Piecewise Linear Signal

Given a piecewise linear signal with additive noise as in Fig. 13.8 upper plot, we wish to estimate the occurrences and parameters of the switches
For that purpose, we use a single signal component \( s^{[1]} \) which consists of occurrences of signal shapes from some class. Here, the signal class comprises all lines, i.e.,

\[
C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \mathbb{R}^2.
\]

Furthermore, \( B_k^{[1]} \in \mathbb{R}^2 \) corresponds to the offset and slope of the added line.

In Fig. 13.8, we present the result of our algorithm. The non-zero values of \( \hat{\sigma}_{U_k}^{2} \) (lower plot, in red) indicates the estimated locations of changes, to be compared with the actual inputs (lower plot, in gray). Despite a high level of noise, the estimated signal component \( \hat{s}^{[1]} \) provides a rather good estimation of the observed signal. The slope and offset of each line switch are not perfectly matched but provide a visually plausible estimate. In addition, the locations of the switches are almost all accurately detected. Only one switch is wrongly detected and is better.
estimated by two switches instead of one. However, with the presence of such high level of noise, this is not a surprise.

**Piecewise Smoothly-Varying Signal**

Given a piecewise smoothly-varying signal with additive noise as in Fig. 13.8 upper plot, we wish to estimate the occurrences and parameters of the switches (i.e., when the signal jumps).

For that purpose, we use a single signal component \( s^{[1]} \) which consists of occurrences of signal shapes from some class. Here, the signal class comprises all constant signals, i.e.,

\[
\begin{align*}
n_1 &= 1 \\
C_1 &= 1 \\
A_1 &= 1 \\
B_1 &= \mathbb{R}.
\end{align*}
\]

Furthermore, \( B_k^{[1]} \in \mathbb{R} \) corresponds to the constant signal value. Even if this model seems simplistic to describe a piecewise smoothly-varying signal, remember that we always have a state noise term via the variance parameter \( \sigma_{E^{[1]}}^2 \), which can be seen as a random walk term. Thus, if the assumption that the signal is piecewise constant is not good enough, the
13.3 Sparse Signal Decomposition in a Signal Shape Domain

estimated variance $\hat{\sigma}_{E[1]}^2$ should be of substantial value to compensate for the model mismatch.

In Fig. 13.9, we present the result of our algorithm. The non-zero values of $\hat{\sigma}_{E[1]}^2$ (lower plot, in red) indicates the estimated locations of changes, to be compared with the actual inputs (lower plot, in gray). Except for one false positive detection, the locations of the switches are all accurately detected. Moreover as expected, the estimated state noise variance is big enough to compensate for the model mismatch. Namely, $\hat{\sigma}_{E[1]}^2 = 6 \cdot 10^{-3}$ allows smooth variations in-between each signal piece, as shown by the estimated signal component $\hat{s}[1]$ (upper plot, red line) which is far from being piecewise constant.

13.3 Sparse Signal Decomposition in a Signal Shape Domain

13.3.1 Problem Setup

Let $g(t), t \in \mathbb{R}$, be a given continuous-time signal shape. We wish to decompose multi-channel discrete time observations $y_k \in \mathbb{R}^M, k \in \{1, \ldots, K\}$ into a linear combination of $P \in \mathbb{N}$ intermediate signal components $\tilde{s}_k^p \in \mathbb{R}, k \in \{1, \ldots, K\}$ where each component is composed of few scaled, time-shifted, and time-dilated (and discretized) versions of $g$. Specifically, we have

$$y_k = \tilde{D} \begin{bmatrix} \tilde{s}_k^1 \\ \vdots \\ \tilde{s}_k^P \end{bmatrix} + Z_k, \quad (13.23)$$

for $k \in \{1, \ldots, K\}$, with $\tilde{D} \in \mathbb{R}^{M \times P}$ and $Z_k \text{iid} \sim \mathcal{N}(0, \sigma_Z^2 I_M)$. The matrix $\tilde{D}$ might be fixed or unknown.

This problem resembles a sparse wavelet decomposition. However, since $g$ can be any signal shape, our problem is more general and also allows the use of a mixing matrix $\tilde{D}$. When confronted with such decomposition problems and assuming that the effect of $\tilde{D}$ has already been taken care of, a straightforward solution is to approximate $g$ with a wavelet [50]. Often, the wavelet is chosen such that its time-shifted and dilated versions form an orthonormal basis of the finite-energy signals. As a result, the discrete-time wavelet transform can be used and
a thresholding of the wavelet coefficients would provide a solution to the given problem. Such a solution faces many problems. Namely, the mixing matrix $\tilde{D}$ is not properly taken into account (one would have to somehow invert the mixing effect) and $g$ is not necessarily well approximated by some wavelet.

We here provide a different solution based on our proposed framework. First, we approximate $g$ with a continuous-time two-sided LSSM signal, i.e.,

$$\hat{g}(t) = \begin{cases} C_p e^{tA_p} s_p & \text{for } t \leq 0 \\ C_f e^{tA_f} s_f & \text{for } t > 0 \end{cases},$$ (13.24)

with left-sided parameters $n_p \in \mathbb{N}$, $C_p \in \mathbb{R}^{1 \times n_p}$, $A_p \in \mathbb{R}^{n_p \times n_p}$, and $s_p \in \mathbb{R}^{n_p}$ and right-sided parameters $n_f \in \mathbb{N}$, $C_f \in \mathbb{R}^{1 \times n_f}$, $A_f \in \mathbb{R}^{n_f \times n_f}$, and $s_f \in \mathbb{R}^{n_f}$. Note that the class of LSSM signals is more diverse than the set of orthogonal wavelets.

Since it is intractable to consider all possible time dilatations $\tau \in \mathbb{R}_+$, we only consider a finite set of time dilatations $\tau \in \\{\tau_q : q \in \{1, \ldots, Q\}\}$. Thus, each intermediate signal component $\tilde{s}^{[p]}$, $p \in \{1, \ldots, P\}$, is further decomposed into a sum of $Q$ signal components $s^{[p,q]}$, i.e.,

$$\tilde{s}^{[p]}_k = \sum_{q=1}^{Q} s^{[p,q]}_k,$$ (13.25)

for $k \in \{1, \ldots, K\}$, and where $s^{[p,q]}$ admits a LSSM representation as in (10.2), which consists of repetitions of the signal shape $\hat{g}(k/\tau_q)$, $k \in \mathbb{Z}$ (cf. Sec. 10.3.1). Hence, the LSSM parameters of $s^{[p,q]}$ are

$$C_{p,q} = \begin{bmatrix} C_p & C_f \end{bmatrix} \in \mathbb{R}^{1 \times n}$$ (13.26)

$$A_{p,q} = \text{diag} \left( e^{-A_p/\tau_q}, e^{A_f/\tau_q} \right) \in \mathbb{R}^{n \times n}$$ (13.27)

$$B_{k}^{[p,q]} = \begin{bmatrix} s_p \\ s_f \end{bmatrix} \in \mathbb{R}^n,$$ (13.28)

with $n = n_p + n_f$ (cf. Sec. 2.4.2 and 4.1.4). Note that an isotropic state noise term is usually added for stability.

Finally, (13.23) expresses as in (10.1) with

$$y_k = D \begin{bmatrix} s^{[1,1]}_k \\ s^{[1,2]}_k \\ \vdots \\ s^{[P,Q]}_k \end{bmatrix} + Z_k,$$ (13.29)
where

\[ D = \hat{D} \text{diag}(\mathds{1}_{1 \times Q}, p \in \{1, \ldots, P\}) \in \mathbb{R}^{M \times (PQ)}. \]  

(13.30)

Note that \( \mathds{1}_{1 \times Q} \) denotes a row vector of ones of size \( Q \) while

\[ \text{diag}(\mathds{1}_{1 \times Q}, p \in \{1, \ldots, P\}) \in \mathbb{R}^{P \times (PQ)} \]

denotes a block diagonal matrix with \( P \) diagonal blocks all being \( \mathds{1}_{1 \times Q} \).

As a result, using a NUV prior for the inputs of each signal component \( s[p, q], p \in \{1, \ldots, P\}, q \in \{1, \ldots, Q\} \), we wish to explain the multi-channel observations \( y_k \in \mathbb{R}^M, k \in \{1, \ldots, K\} \) with a joint LSSM of order \( n \cdot M \cdot Q \) as in (10.5). All input parameters and unknown LSSM parameters are estimated by maximum likelihood using, for instance, an EM algorithm.

In the present description of the solution, all the LSSM parameters except for the state noise variances are fixed. Nevertheless, our framework is flexible enough to incorporate more unknown parameters such as the discrete set of time dilatations \( \{\tau_q : q \in \{1, \ldots, Q\}\} \). However, we here do not exploit this possibility.

### 13.3.2 Inferring Electric Dipole Movements From Multi-Channel Electrode Measurements

We now present an example application which actually motivated the definition of the problem setup described in Sec. 13.3. In this application, we want to infer straight-line movements of several electric dipoles from multi-channel measurements. Compared to the application described in Sec. 8.3, the voltage signals produced by one dipole can overlap with voltage signals produced by other dipoles; they are not nicely separated in time anymore. Furthermore, the multi-channel sensor is not assumed to be of small dimensions. Consequently, a noticeable delay between the signals observed in each measurement channel is now to be handled.

**Voltage Signals Produced by a Moving Dipole**

Consider a fixed electrode geometry consisting of \( M + 1 \) electrode points \( H_1, \ldots, H_{M+1} \). Let \( O \) denote the center of coordinate that is chosen to be a reference point of the electrode. Assume that an electric dipole of dipole moment \( \mathbf{p} \in \mathbb{R}^3 \) is moving uniformly on a straight line. Let \( P_t, t \in \mathbb{R}, \) denote the instantaneous position of the dipole. At a given time
$t \in \mathbb{R}$, the potential $\Phi_H(t)$ created by the dipole at an electrode point $H \in \{H_1, \ldots, H_{M+1}\}$ is

$$\Phi_H(t) = \frac{\langle p, \overrightarrow{P_H} \rangle}{\|P_H\|^3}, \quad (13.31)$$

where $p$ also encompasses the constant factor $\frac{1}{4\pi\epsilon}$ which appears in the dipole equation (8.67).

For a uniform straight-line movement of a dipole, we further have

$$\overrightarrow{P_H} = \overrightarrow{P_0} - tv, \quad (13.32)$$

where $v = \nu \overrightarrow{d_v}$ ($v \in \mathbb{R}_+, \|\overrightarrow{d_v}\| = 1$) is the speed vector. Thus, plugging (13.32) in (13.31), we have

$$\Phi_H(t) = \frac{\langle p, \overrightarrow{P_0} \rangle - tv \langle p, \overrightarrow{d_v} \rangle}{\left(\|\overrightarrow{P_0}\|^2 - 2tv\langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle + t^2v^2\right)^{3/2}}. \quad (13.33)$$

Manipulating the expression appearing in the denominator of this fraction, we obtain

$$\|\overrightarrow{P_0}\|^2 - 2tv\langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle + t^2v^2$$

$$= \left(\|\overrightarrow{P_0}\|^2 - \langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle^2\right) \left(1 + \left(\frac{tv - \langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle}{\sqrt{\|\overrightarrow{P_0}\|^2 - \langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle^2}}\right)^2\right)$$

$$= \frac{1}{\alpha_H} \left(1 + \left(\frac{t - t_H}{\tau_H}\right)^2\right), \quad (13.34)$$

with

$$\alpha_H = \frac{1}{\|\overrightarrow{P_0}\|^2 - \langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle^2} \quad (13.36)$$

$$\tau_H = \frac{1}{v} \sqrt{\|\overrightarrow{P_0}\|^2 - \langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle^2} \quad (13.37)$$

$$t_H = \frac{\langle \overrightarrow{P_0}, \overrightarrow{d_v} \rangle}{v}. \quad (13.38)$$

Rearranging (13.33), we finally get

$$\Phi_H(t) = \alpha_H \frac{\beta_H - \langle p, \overrightarrow{d_v} \rangle \left(\frac{t - t_H}{\tau_H}\right)}{\left(1 + \left(\frac{t - t_H}{\tau_H}\right)^2\right)^{3/2}}, \quad (13.39)$$
with

\[
\beta_H = \frac{\langle P_0 \vec{H}, \mathbf{p} - \langle \mathbf{p}, \mathbf{d}_v \rangle \mathbf{d}_v \rangle}{\sqrt{\|P_0 \vec{H}\|^2 - \langle P_0 \vec{H}, \mathbf{d}_v \rangle^2}}. \tag{13.40}
\]

We now confuse index \(m\) with index point \(H_m\). Thus, each potential \(\Phi_m\) at electrode point \(H_m\) produced by a single moving dipole is a linear combination of the two functions

\[
g(x) = \frac{x}{(1 + x^2)^{\frac{3}{2}}} \tag{13.41}
\]

\[
h(x) = \frac{1}{(1 + x^2)^{\frac{3}{2}}} \tag{13.42}
\]

with a time shift of \(t_m\) and a time dilatation of \(\tau_m\), that is to say,

\[
\Phi_m(t) = \alpha_m \left(-\langle \mathbf{p}, \mathbf{d}_v \rangle g\left(\frac{t - t_m}{\tau_m}\right) + \beta_m h\left(\frac{t - t_m}{\tau_m}\right)\right). \tag{13.43}
\]

For simplicity, we assume that the dipole moment is aligned with its movement direction (i.e., \(\mathbf{p} \propto \mathbf{d}_v\)). It follows that \(\beta_m = 0\), for all \(m \in \{1, \ldots, M+1\}\). Denoting \(\lambda_m = -\alpha_m \langle \mathbf{p}, \mathbf{d}_v \rangle\), the potential \(\Phi_m\) further simplifies into

\[
\Phi_m(t) = \lambda_m g\left(\frac{t - t_m}{\tau_m}\right). \tag{13.44}
\]

Thus, each potential \(\Phi_m\) at \(H_m\) is a version of the function \(g\) with a scale of \(\lambda_m\), a time shift of \(t_m\), and a time scale of \(\tau_m\). The signal shape \(g\) is plotted in Fig. 13.10, blue line.

Furthermore, what is measured by the electrodes are the voltage signals

\[
V_m(t) = \Phi_{m+1}(t) - \Phi_m(t), \tag{13.45}
\]

for \(m \in \{1, \ldots, M\}\), at discrete time steps.

**An Example of Measured Voltage Signals**

For illustrative purposes, we simulate the voltage signals produced by a single moving dipole and recorded with a 3-channel inline electrode as in Fig. 13.11 with electrode points \(\{H_1, H_2, H_3, H_4\}\) separated by 1.5 cm from each other. We use \(H_1\) as the center of the coordinate system.
Figure 13.10 – Signal shape $g$ and estimated LSSM signal $\hat{g}$.

Figure 13.11 – Inline electrode and moving electric dipole.

The electric dipole has a minimum distance of $r = 5$ cm to the electrode, a speed of $v = 4$ m/s, directions

$$d_r = \begin{bmatrix} -\sin(2\pi/3) & \cos(2\pi/3) & 0 \end{bmatrix}^T$$  \hspace{1cm} (13.46)
$$d_v = \begin{bmatrix} \cos(2\pi/3) & \sin(2\pi/3) & 0 \end{bmatrix}^T,$$  \hspace{1cm} (13.47)

and a dipole moment $p = 10^{-2}$ V/m$^2$ with $d_p = d_v$.

For this configuration, the signal parameters are given in Table 13.1 and the potentials at each electrode point $H_1, \ldots, H_4$ and the voltages $V_m, m \in \{1, \ldots, 3\}$, measured between consecutive electrode points are plotted in Fig. 13.12.
Table 13.1 – Parameters of the 3-channel voltage signals produced by a given dipole movement.

<table>
<thead>
<tr>
<th>m</th>
<th>Shift $t_m$ (ms)</th>
<th>Dilatation $\tau_m$ (ms)</th>
<th>Scale $\lambda_m$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>12.5</td>
<td>$-4.0$</td>
</tr>
<tr>
<td>2</td>
<td>3.247</td>
<td>10.625</td>
<td>$-5.536$</td>
</tr>
<tr>
<td>3</td>
<td>6.495</td>
<td>8.75</td>
<td>$-8.1632$</td>
</tr>
<tr>
<td>4</td>
<td>9.743</td>
<td>6.875</td>
<td>$-13.2231$</td>
</tr>
</tbody>
</table>

LSSM Modeling

The function $g$ can be well-approximated with a 2nd-order LSSM

$$\hat{g}(x) = Ke^{-\alpha|x|} \sin(\omega x),$$

with $K = 104.6$, $\alpha = 1.414$, and $\omega = 0.01414$.

Thus, the LSSM parameters of this estimated signal, as described in (13.24), are

$$A_p = -\alpha I_2 - \omega \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

$$A_f = -\alpha I_2 + \omega \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

$$C_p = C_f = \begin{bmatrix} 0 & K \end{bmatrix},$$

$$s_p = s_f = \begin{bmatrix} 1 & 0 \end{bmatrix}^T,$$

such that for $t \leq 0$,

$$e^{||t||A_p} = e^{\alpha t} R(\omega t),$$

and for $t \geq 0$,

$$e^{tA_f} = e^{-\alpha t} R(\omega t).$$

In Fig. 13.10 we observe that the LSSM signal shape $\hat{g}$ (red dashed line) nicely fits the original shape $g$ (blue line). If necessary, the fit can be improved using a higher-order LSSM.

Algorithm Description

Given multi-channel discrete-time voltage measurements $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, M\}$, we wish to detect straight-line passes of electric dipoles (with
dipole moments aligned with their movement directions). Since a single dipole moment produces a potential with a signal shape $g$ as in (13.41) with a time shift, an amplitude scale, and a time dilatation at each electrode point, several dipoles produce superpositions of such potentials.

The problem we have just described is quite hard to solve. Indeed, many parameters to be estimated come to light: the number of dipole movements and the parameters of each dipole movement in each channel.

Thus, we modify the original problem as follows. We wish to decompose the $M$-channel measurements $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, M\}$ into $P = M + 1$ intermediate signal components (i.e., the $M + 1$ potentials) where each component is composed of a few scaled, time-shifted, and time-dilated (and discretized) versions of $g$. Note that from the solution obtained by solving the modified problem, further inference needs to be done to provide an answer to the original problem. Nevertheless, the solution obtained by solving the modified problem drastically reduces the complexity of the original problem.

The modified problem is an instance of the problem described in Sec. 13.3.1 with $P = M + 1$ and where the matrix $\tilde{D}$ in (13.23) is

$$
\tilde{D} = \begin{bmatrix}
-1 & 1 \\
.& & \ddots & \cdots \\
.& & & 1 \\
-1 & 1 \\
\end{bmatrix} \in \mathbb{R}^{M \times (M+1)}. 
$$

Note that $\tilde{D}$ is a singular matrix.

Using the LSSM model of $g$ described in (13.48) and a finite set of time dilatations $\tau \in \{\tau_q : q \in \{1, \ldots, Q\}\}$, we can run our learning algorithm that is described in Sec. 13.3.1. In this application, the intermediate signal components $\tilde{s}^{[p]}$, $p \in \{1, \ldots, P\}$ correspond to the potentials. Moreover, a non-zero value of the input variance $\sigma_{U_k^{[p,q]}}$ for some $p \in \{1, \ldots, P\}$ and $q \in \{1, \ldots, Q\}$ would indicate that the signal shape $g$ shifted by a time index $k$ and time-dilated by $\tau_q$ occurs in the $p^{th}$ intermediate signal component $\tilde{s}^{[p]}$ (i.e., the $p^{th}$ potential signal).

**Simulation Results**

In Fig. 13.13 and 13.14, we present simulation results of our algorithm when decomposing voltage signals produced by a single dipole movement. Note that $\hat{\Phi}_p$, $p \in \{1, 2, 3, 4\}$, simply stands for the estimated intermediate signal component $\tilde{s}^{[p]}$. 


In the first experiment, we use $M = 2$ voltage signals and we select the time dilatations to exactly match the set of time dilatations produced by the dipole movement. In this example, the 3 time dilatations are
\begin{align*}
\tau_1 &= 25.0 \quad (13.56) \\
\tau_2 &= 28.9 \quad (13.57) \\
\tau_3 &= 38.3, \quad (13.58)
\end{align*}
in per-sample unit. Nonetheless, since all three dilatations can be used by each potential signal, our algorithm still needs to estimate the three potential signals and, in particular, assign one or several time dilations to the corresponding potential signals.

In Fig. 13.13, we observe that our algorithm has no difficulty to correctly assign the potential signals to its right time dilatation. Indeed, for each potential signal $\Phi_m$, $m \in \{1, 2, 3\}$, the estimated potential $\hat{\Phi}_m$ fits to the original one and only uses $\tau_m$ to explain the potential signal; only the elements of $\hat{\sigma}^2_{U_k[m,m]}$ are used while the remaining input variances $\hat{\sigma}^2_{U_k[m,q]}$, $q \neq m$ are all estimated to be zero (not appearing in Fig. 13.13 since all zero). Moreover, the non-zero input variances almost perfectly coincide with the correct time shifts $t_m$, $m \in \{1, 2, 3\}$. In addition, the voltage signals are well estimated, as seen in the lower plot of Fig. 13.13.

In the second experiment, we use $M = 3$ voltage signals but we select only three time dilatations which do not match the exact ones produced by the dipole movement, unlike in the first experiment. We choose the 3 time dilatations as
\begin{align*}
\tau_1 &= 23.7 \quad (13.59) \\
\tau_2 &= 32.3 \quad (13.60) \\
\tau_3 &= 50.0, \quad (13.61)
\end{align*}
whereas the correct set of time dilatations is
\[\{25.0, 28.9, 38.3, 50.1\}. \quad (13.62)\]

As shown in Fig. 13.14, the result of our algorithm is reasonable but not entirely satisfactory. Even if the voltage signals are accurately estimated, the potential signals are not so well estimated. This behavior was expected since the set of allowed time dilatations does not match the right one and one time dilatation is missing. As far as the input variance estimation is concerned, the algorithm is still able to assign the potential
signals to its closest time dilatation in the allowed set. Unfortunately, there is an extra nonzero input variance \( \hat{\sigma}_{U_k[2,3]} \) that is used to explain \( \phi_2 \) and also, the non-zero input variance \( \hat{\sigma}_{U_k[4,3]} \) triggers later than expected. Moreover, with a higher level of noise on the voltages, the algorithm has issues to converge to the desired solution.

**Discussion and Conclusion**

The problem of inferring straight-line movements of several electric dipoles from multi-channel measurements is quite hard to solve and many decent algorithms which intend to solve it would be of high computational complexity since plenty of time shifts, time dilatations, and amplitude scales need to be considered.

Despite the mixed results, our algorithm seems promising for providing a reasonable solution while being of affordable computational complexity. Recall that since the mixing matrix \( \tilde{D} \) is singular, the problem of explaining voltage signals by modeling potential signals is ill-posed if no sparsity assumption is introduced. This makes this specific application quite sensitive since many local optimum solutions exists. Although our approach includes a sparsity assumption, our algorithm is not prevented from converging to a local optimum solution. Thus, the mixed results are both due to the nature of the problem and the actual proposed algorithm.

However, including more constraints in the input domain of the LSSM or in the LSSM parameters would be quite beneficial. Indeed, the knowledge about the geometry of the multi-channel sensor is not included in our method despite the fact that we exactly know how amplitudes, time shifts, and time dilatations are related (see (13.37)–(13.38)). In particular, the observation that a single dipole movement creates amplitude scales of the same sign in all measurement channels could be taken into account. Ideas on how to include more constraints in the input domain are presented in Chapter 14.

To conclude, this challenging application needs further investigation even if our method provides useful insights.
Figure 13.12 – Voltage and potential signals created by a moving dipole.
Figure 13.13 – Observed and estimated voltage and potential signals created by a moving dipole (with a matched set of time dilatations). The input variances $\hat{\sigma}_{U_k^{[p,q]}}$ for $(p, q) \in \{1, 2, 3\}^2, p \neq q$ are all zeros and thus, not displayed.
Figure 13.14 – Observed and estimated voltage and potential signals created by a moving dipole (without a matched set of time dilatations). The input variances $\hat{\sigma}_{U[p,q]}$ for $p \in \{1, 2, 3, 4\}$, $q \in \{1, 2, 3\}$ which are not displayed are all zero.
Chapter 14

Hierarchical Modeling and Processing

Up until now, the main purpose of Part III was to learn sparse signal decompositions using a statistical model $p(y|u, \theta)$ as described in Sec. 10.2, which essentially boils down to estimating the LSSM parameters $\theta$ and sparse inputs $u$. Sparsity has been encoded by modeling inputs with independent zero-mean Gaussian variables with unknown variances $\sigma_u^2$. Thus, no sparsity structure in the input domain of the LSSM has been considered so far.

However, the input domain in which a signal is sparsely represented might also exhibit some characteristic features that we wish to learn. For instance, we would like to know if some signal shapes appear in synchrony either over time or over several channels. If it is the case, a sparsity pattern would appear in the input domain of a sparse LSSM representation and further processing would be required to extract it.

In this chapter, we outline a few ideas on how to extract or handle additional structures in the input domain of a sparse LSSM representation. Note that most of these ideas can be used in a hierarchical way hoping that complex structural information can be encoded with several simple modeling layers.
14.1 Straightforward Hierarchical Processing

Assume that we have learned a sparse signal decomposition of a given multi-channel signal \( y_k \in \mathbb{R}^M, k \in \{1, \ldots, K\} \), using the method described in Chapter 12. This decomposition is characterized by the estimated LSSM parameters \( \hat{\theta} \) and the estimated input variances \( \hat{\sigma}^2_U \) (or alternatively, the posterior input means \( m_U \)). It results that the input variances \( \hat{\sigma}^2_{U(j)} \), \( k \in \{1, \ldots, K\}, j \in \{1, \ldots, J\} \), where \( J \) is the number of allowed inputs per time index, is a sparse representation of the original signal, which indicates if something happens at any time index and, on an abstract level, what happens if something occurs.

Thus, a straightforward way to deal with additional input structures would be to directly process the sparse multi-channel representation \( \hat{\sigma}^2_U \) (or \( m_U \)) in order to recognize patterns. For this task, we could reuse our method for learning sparse signal decompositions by considering the variances \( \hat{\sigma}^2_U \) as observations. Alternatively, many pattern recognition algorithms [8, 29, 72], such as neural networks, can be used. We would also like to point out the approach described in [54], which focuses on efficiently processing information that is represented with pulses.

In any case, we see that this technique for processing information can be applied hierarchically and can potentially extract complex structural information.

14.2 A Modeling Layer for the Inputs

Assume that we are given a statistical model \( p(y|u, \theta) \) as described in Sec. 10.2, which encodes a signal decomposition. Both \( \theta \) and \( u \) are unknown. Now, we still assume that \( u \) is sparse but we further assume that there exists a representation of \( u \) that is even sparser than \( u \) itself. In other words, we are also looking for a prior statistical model \( p(u|\tilde{\theta}, v) \) where \( \tilde{\theta} \) are some model parameters and where \( v \) is assumed to be sparser than \( u \). Then, \( v \) can be modeled with NUV terms and is associated with the unknown variances \( \sigma^2_V \). Thus, the sparse signal decomposition problem amounts to estimating the parameters \( \theta, \tilde{\theta} \), and \( \sigma^2_V \). Note that this estimation problem usually requires the use of iterative algorithms and approximations.

A particular case is when \( p(u|\tilde{\theta}, v) \) is described with a LSSM as in Sec. 10.2. In fact, in this case, \( p(y|v, \theta, \tilde{\theta}) \) also has a LSSM representation,
as described in Sec. 2.3. Thus, a similar algorithm as in Chapter 12 can be used to estimate all unknown parameters. Since not so many LSSMs produce a sparse output when fed with sparse inputs, we do not necessarily expect \( p(u|\tilde{\theta}, v) \) to be a good generative model but rather a good discriminative model.

### 14.3 A Modeling Layer for the Input Variances

#### 14.3.1 General Model

Assume that we are given a statistical model \( p(y|\theta, \sigma^2_U) \) as described in Sec. 12.1, which encodes a signal decomposition. Both \( \theta \) and \( \sigma^2_U \) are unknown.

Additional input structures can be added via a statistical model \( p(\sigma^2_U|\tilde{\theta}, v) \) where \( \tilde{\theta} \) are some model parameters and where \( v \) is assumed to be sparse. Then, \( v \) can be modeled with NUV terms and is associated with the unknown variances \( \sigma^2_V \). Thus, the signal decomposition problem boils down to estimating the parameters \( \theta, \tilde{\theta}, \) and \( \sigma^2_V \). Note that this estimation problem usually requires the use of iterative algorithms and approximations.

#### 14.3.2 Independent Model

In the special case where \( p(\sigma^2_U|\tilde{\theta}, v) \) factorizes as

\[
p(\sigma^2_U|\tilde{\theta}, v) = \prod_{k=1}^{K} \prod_{j=1}^{J} \psi_j \left( \sigma^2_{U_{(j)}} \right),
\]

where \( \psi_j \) are some functions, we essentially retrieve a GSM model as described in Sec. 11.2. Thus, instead of modeling variances with an improper prior distribution, we can encode prior information on the values of the variances via the functions \( \psi_j, j \in \{1, \ldots, J\} \).

In addition, this extension is easily included into the EM learning algorithm described in Chapter 12 (see [56]).
14.3.3 Equality Constraints

Equality Over Time

For a given input dimension \( j \in \{1, \ldots, J\} \), if we want to enforce that either all inputs trigger (i.e., are non-zero) over time or none of them, we can simply constrain all the \( \sigma_{U^{(j)}}^{2} \) for \( k \in \{1, \ldots, K\} \) to be equal. This modification is easily handled in the EM algorithm of Chapter 12 by replacing (12.20) with

\[
\hat{\sigma}_{U^{(j)}}^{2} = \frac{1}{K} \sum_{k'=1}^{K} \mathbb{E} \left[ \left( U_{k'}^{(j)} \right)^{2} \right], \tag{14.2}
\]

for \( k \in \{1, \ldots, K\} \).

Equality Over Input Dimensions

Let \( J \) be a subset of \( \{1, \ldots, J\} \). Suppose that for each time index \( k \in \{1, \ldots, K\} \), we want to enforce that either all inputs in the dimensions \( j \in J \) trigger (i.e., are non-zero) or none of them. Then, we can simply constrain all the \( \sigma_{U^{(j)}}^{2} \) for \( j \in J \) to be equal. This modification is easily handled in the EM algorithm of Chapter 12 by replacing (12.20) with

\[
\hat{\sigma}_{U^{(j)}}^{2} = \frac{1}{|J|} \sum_{j' \in J} \mathbb{E} \left[ \left( U_{k}^{(j')} \right)^{2} \right], \tag{14.3}
\]

for \( k \in \{1, \ldots, K\} \) and \( j \in J \).

Equality Over Input Dimensions and Time

If we want to enforce that either all inputs in the dimensions \( j \in J \) trigger (i.e., are non-zero) over time or none of them, we can simply constrain all the \( \sigma_{U^{(j)}}^{2} \) for \( k \in \{1, \ldots, K\} \) and \( j \in J \) to be equal. Then, (12.20) is replaced with

\[
\hat{\sigma}_{U^{(j)}}^{2} = \frac{1}{|J|} \sum_{j' \in J} \sum_{k'=1}^{K} \mathbb{E} \left[ \left( U_{k'}^{(j')} \right)^{2} \right], \tag{14.4}
\]

for \( k \in \{1, \ldots, K\} \) and \( j \in J \).
14.3.4 Individual Input Constraints

In some cases we would like to constrain the inputs $u$ to some set $U \subset \mathbb{R}^{KJ}$. For instance, having only positive inputs might be a requirement. Unfortunately, in the current formulation of the learning problem, i.e., minimizing the function (cf. (11.12))

$$J(\theta, \sigma_u^2) = y^T(V_Y + H\Sigma_U H^T)^{-1}y + \ln|V_Y + H\Sigma_U H^T|,$$  \hspace{1cm} (14.5)

we cannot directly achieve such behavior as explained in Sec.11.3.

Nonetheless, since the final MAP estimation of $u$ is given by (11.9), constraining the inputs can be achieved by minimizing $J(\theta, \sigma_u^2)$ under the constraint that the estimated $u$ belongs to $U$, i.e.,

$$\Sigma_U H^T(V_Y + H\Sigma_U H^T)^{-1}y \in U.$$ \hspace{1cm} (14.6)

Despite a nice formulation, this constrained minimization problem is not easily handled in practice due to a constraint which does not usually split (element-wise) and which depends on both $\theta$ and $\sigma_u^2$.

In case $U$ constrains each element of $u$ individually, i.e., $u_{(j)^{(k)}} \in \tilde{U}$ the constraint (14.6) still does not split. However, when performing Gaussian message passing, the value of the posterior mean $m_{U_{(j)^{(k)}}}$ is easily obtained using (4.40) or (12.76). Instead of globally imposing the constraint (14.6), we could also enforce approximate local constraints $m_{U_{(j)^{(k)}}} \in \tilde{U}$. For instance, such local constraints could be naively included in some way within the EM algorithm but without any guarantee.

Note that in [80] which only considers the estimation of $\sigma_u^2$ (and not $\theta$), an alternative optimization algorithm is proposed which can also ensure non-negativity of the inputs $u$. 
Chapter 15

Conclusion

In this thesis, we have strongly advocated the use of LSSM in order to learn sparse signal decompositions and also detect and estimate events. While LSSMs bring powerful modeling capabilities, they also provide a structured, compact representation which makes signal separation and learning possible via estimation algorithms.

When events are sufficiently separated in time, the method for event detection and estimation described in Part II is computationally efficient and ready for a nearly online implementation. Moreover, it can tackle various detection and estimation problems even in the presence of an unknown interference signal, as reflected by the numerous example applications we have provided.

The general method proposed in Part III for learning sparse signal decompositions is extremely versatile and provides a unifying view and a joint solution for the three problems: separating signal components, finding sparse signal representations, and learning signal spaces in which the signal components can be sparsely represented. Since sparsity is encoded with the use of zero-mean Gaussian variables with unknown variances, efficient learning algorithms such as EM and Gaussian message passing are available and makes the proposed approach practical. In addition, both the experimental and simulated results are promising and suggest that our proposed approach can solve a plethora of challenging practical applications involving learning sparse signal decompositions.
Appendices
“The trouble with the world is that the stupid are cocksure and the intelligent are full of doubt.”

Bertrand Russell
Appendix A

Matrix Formulas

In all the following formulas, $A$, $B$, $C$, and $D$ are matrices. Furthermore, we assume that matrix multiplications, matrix inverses, and matrix determinants are well defined accordingly.

Formula 1 (Kronecker Product and Matrix Multiplication).

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad (A.1)$$

Formula 2 (Vectorization and Matrix Multiplication).

$$\text{vec}(ABC) = (C^T \otimes A) \text{vec}(B) \quad (A.2)$$

Formula 3 (Trace and Vectorization).

$$\text{tr}(A^TBCD^T) = \text{vec}(A)^T(D \otimes B)\text{vec}(C) \quad (A.3)$$

Formula 4 (Matrix Inversion Lemma).

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (A.4)$$

Formula 5 (Matrix Determinant Lemma).

$$|A + BCD| = |C^{-1} + DA^{-1}B| \cdot |C| \cdot |A| \quad (A.5)$$

Formula 6 (Block Matrix Inversion).

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} - A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} \end{bmatrix} \quad (A.6)$$
Formula 7 (Block Matrix Inversion (version 2)).

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^{-1} = \begin{bmatrix}
(A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\
-D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1}
\end{bmatrix}
\] (A.7)

Formula 8 (Inverse Matrix Derivative).

\[dA^{-1} = -A^{-1}(dA)A^{-1}\] (A.8)

Formula 9 (Log-Determinant Derivative).

\[d\ln|A| = \text{tr}(A^{-1}(dA))\] (A.9)

Formula 10 (Matrix Trace Inequality [21]). Let \(\Lambda, \Gamma \in \mathbb{R}^{n \times n}\) be diagonal matrices with diagonal elements in decreasing order. For all orthogonal matrices \(U \in O_n(\mathbb{R})\),

\[\text{tr}(U\Lambda U^T \Gamma) \leq \text{tr}(\Lambda \Gamma),\] (A.10)

and equality holds with \(U = I_n\).

Formula 11 (Another Matrix Trace Inequality [21]). Let \(\Lambda, \Gamma \in \mathbb{R}^{n \times n}\) be diagonal matrices, one with diagonal elements in decreasing order and the other one in increasing order. For all orthogonal matrices \(U \in O_n(\mathbb{R})\),

\[\text{tr}(U\Lambda U^T \Gamma) \geq \text{tr}(\Lambda \Gamma),\] (A.11)

and equality holds with \(U = I_n\).

Formula 12 (Gaussian Conditioning). If

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}
m_X \\
m_Y
\end{bmatrix}, \begin{bmatrix}
V_{X} & V_{X,Y} \\
V_{X,Y}^T & V_{Y}
\end{bmatrix}\right),
\] (A.12)

then, for any \(y\),

\[X | Y = y \sim \mathcal{N}(m_X + V_{X,Y}V_Y^{-1}(y - m_Y), V_X - V_{X,Y}V_Y^{-1}V_{X,Y}^T).\] (A.13)
Appendix B

LSSM-Weighted Squared Error and Covariance Estimation

Given a multi-channel discrete-time signal $y_1, \ldots, y_K \in \mathbb{R}^M$, $M \in \mathbb{N}$, and a $M$-channel LSSM signal $f(C)$ with parameters $\{C, A, s\}$ and unknown $C$, let $\epsilon_k(C) \in \mathbb{R}^{M \times M}$ be the LSSM-weighted sample covariance matrix

\[
\epsilon_k(C) = \frac{1}{\nu_k} \left( \epsilon_k(C_p) + \epsilon_k(C_f) \right),
\]

with $\nu_k = \sum_{i=1}^{K} w_i - k$,

\[
\epsilon_k(C_p) = \sum_{i=1}^{k} w_i - k \left( y_i - C_p A_p^{k-i} s_p \right) \left( y_i - C_p A_p^{k-i} s_p \right)^T \quad (B.3)
\]

\[
\epsilon_k(C_f) = \sum_{i=k+1}^{K} w_i - k \left( y_i - C_f A_f^{i-k} s_f \right) \left( y_i - C_f A_f^{i-k} s_f \right)^T, \quad (B.4)
\]

where $w_i - k$ is a (non-negative) LSSM signal with given parameters $\{C_w, A_w, s_w\}$ and shifted by a time $k$. 
Using the quantities defined in Sec. 6.2.2, we have the relation
\[ \epsilon_k(C) = \frac{1}{\nu_k} (E_k - C\xi_k - \xi_k^TC^T + CW_kC^T), \] (B.5)
where
\[ E_k = \sum_{i=1}^{K} w_{i-k}y_iy_i^T. \] (B.6)
Since \( E_k, \xi_k, \) and \( W_k \) can be recursively computed (according to Proposition 9), \( \epsilon_k(C) \) can also be recursively computed.

We now define a general local likelihood as in (7.8) but with a general covariance matrix \( V = W^{-1} \). Thus, we obtain
\[ \hat{p}(y_1, \ldots, y_K; C, W) = \prod_{i=1}^{K} \mathcal{N}(y_i : f_{i-k}(C), V_{w_{i-k}}^{w_{i-k}}) \] (B.7)
\[ = \left( \frac{|W|}{(2\pi)^M} \right)^{\frac{\nu_k}{2}} \exp \left( -\frac{\nu_k}{2} \text{tr} (W\epsilon_k(C)) \right). \] (B.8)
Maximizing this local likelihood function is equivalent of minimizing
\[ L(C, W) = \text{tr} (W\epsilon_k(C)) - \ln |W|. \] (B.9)
Assuming that \( W \) is unknown but structured, we now investigate the minimization problem over \( C \in \mathcal{C} \) obtained by finding
\[ \hat{C} = \arg\max_{C \in \mathcal{C}} \max_{W} \hat{p}(y_1, \ldots, y_K; C, W) \] (B.10)
\[ = \arg\min_{C \in \mathcal{C}} \min_{W} L(C, W). \] (B.11)
Consequently, constraining \( W \) is equivalent to regularizing the estimation of \( C \in \mathcal{C} \).

### B.1 Isotropic Uncorrelated Noise

When \( W = \sigma_Z^2 I_M \), we obtain
\[ \hat{\sigma}_Z^2 = \frac{1}{M} \text{tr}(\epsilon_k(C)) \] (B.12)
and thus,
\[
\hat{C} = \arg\min_{C \in \mathcal{C}} \text{tr}(\epsilon_k(C)).
\] (B.13)

This case is exactly the one described in Sec. 7.1.1 since
\[
\nu_k \text{tr}(\epsilon_k(C)) = J_k(C),
\] (B.14)

where \( J_k(C) \) is the LSSM-weighted squared error (6.87). Thus, minimizing \( J_k(C) \) is equivalent to minimizing the trace of the sample covariance matrix \( \epsilon_k(C) \), that is to say, the sum of the per-channel squared errors.

### B.2 Uncorrelated Noise

When \( W_Z = \text{diag}(\sigma_1^{-2}, \ldots, \sigma_M^{-2}) \), we obtain
\[
L(C, W_Z) = \sum_{m=1}^{M} \frac{1}{\sigma_m^2} \{\epsilon_k(C)\}_{m,m} + \ln \sigma_m^2.
\] (B.15)

Thus, we get
\[
\hat{\sigma}_m^2 = \{\epsilon_k(C)\}_{m,m},
\] (B.16)

for \( m \in \{1, \ldots, M\} \), and
\[
\hat{C} = \arg\min_{C \in \mathcal{C}} \prod_{m=1}^{M} \{\epsilon_k(C)\}_{m,m}.
\] (B.17)

In this case, \( \hat{C} \) minimizes the product of the diagonal elements of the sample covariance matrix \( \epsilon_k(C) \), that is to say, the product of the per-channel squared errors.

### B.3 Fully Correlated Noise

When \( W_Z \) is any symmetric positive definite matrix, we obtain that
\[
\hat{W}_Z = \arg\min_{W_Z \in \mathbb{S}_+} \text{tr} (W_Z \epsilon_k(C)) - \ln |W_Z|
\] (B.18)

\[
= \epsilon_k(C)^{-1},
\] (B.19)

assuming that \( \epsilon_k(C) \) is invertible. Consequently, we get
\[
\hat{C} = \arg\min_{C \in \mathcal{C}} |\epsilon_k(C)|.
\] (B.20)

Thus, \( \hat{C} \) minimizes the determinant of the sample covariance matrix.
Appendix C

Cross-Covariance Computations in LSSM

Referring to Fig. 4.1 the joint posterior density \( p(x'_{k-1}, u_k, x_k | y, \theta) \) writes as

\[
p(x'_{k-1}, u_k, x_k | y, \theta) \\
\propto \mathcal{N}(x'_{k-1} : \bar{m}_{X'_{k-1}}, \bar{V}_{X'_{k-1}}) \mathcal{N}(u_k : 0, \Sigma_U) \\
\cdot \mathcal{N}(x_k : Ax'_{k-1} + B_k u_k, V_E) \mathcal{N}(x_k : \bar{m}_{X_k}, \bar{V}_{X_k}),
\]

(C.1)

where \( \propto \) means equality up to a scale factor independent of \( x'_{k-1}, u_k, \) and \( x_k \). Note that we here use the random vector \( X'_{k-1} \). Indeed, since \( X_{k-1} \) and \( X'_{k-1} \) are linked via an equality node, we can replace \( X_{k-1} \) by \( X'_{k-1} \) when computing any (posterior) expectation involving \( X_{k-1} \).

C.1 Cross-Covariance Between \( X_{k-1} \) and \( X_k \)

By integrating (C.1) over \( u_k \), we obtain

\[
p(x'_{k-1}, x_k | y, \theta) \propto \mathcal{N}(x'_{k-1} : \bar{m}_{X'_{k-1}}, \bar{V}_{X'_{k-1}}) \mathcal{N}(x_k : \bar{m}_{X_k}, \bar{V}_{X_k}) \\
\cdot \mathcal{N}(x_k : Ax'_{k-1}, V_E + B_k \Sigma_U B_k^T).
\]

(C.2)

Thus, denoting

\[
V_N = V_E + B_k \Sigma_U B_k^T,
\]

(C.3)
the precision matrix of the joint density (C.2) is

\[
W_{(X'_{k-1}, X_k)} = \begin{bmatrix}
\hat{V}_{X_{k-1}}^{-1} + A^T V_N^{-1} A & -A^T V_N^{-1} \\
-V_N^{-1} A & \hat{V}_{X_k}^{-1} + V_N^{-1}
\end{bmatrix}.
\] (C.4)

Then, the cross-covariance matrix \( V_{X'_{k-1}, X_k} \) corresponds to the upper right block of \( W_{(X'_{k-1}, X_k)}^{-1} \). We now use the block-wise inversion formula (A.6) to compute this cross-covariance matrix. Using the notation in (A.6), we get

\[
“(D - CA^{-1}B)^{-1}”
\]

\[
= (\hat{V}_{X_k}^{-1} + V_N^{-1} - V_N^{-1} A(\hat{V}_{X_{k-1}}^{-1} + A^T V_N^{-1} A)^{-1} A^T V_N^{-1})^{-1}
\] (C.5)

\[
= (\hat{V}_{X_k}^{-1} + (V_N + A\hat{V}_{X_{k-1}} A^T)^{-1})^{-1}
\] (C.6)

\[
= (V_N + A\hat{V}_{X_{k-1}} A^T)(\hat{V}_{X_k} + V_N + A\hat{V}_{X_{k-1}} A^T)^{-1}\hat{V}_{X_k}
\] (C.7)

\[
= (V_N + A\hat{V}_{X_{k-1}} A^T)\tilde{W}_{X_k} \hat{V}_{X_k}.
\] (C.8)

Moreover, we also have

\[
“- A^{-1}B”
\]

\[
= (\hat{V}_{X_{k-1}}^{-1} + A^T V_N^{-1} A)^{-1} A^T V_N^{-1}
\] (C.9)

\[
= (\hat{V}_{X_{k-1}}' - \hat{V}_{X_{k-1}} A^T (V_N + A\hat{V}_{X_{k-1}} A^T)^{-1} A\hat{V}_{X_{k-1}}') A^T V_N^{-1}
\] (C.10)

\[
= \hat{V}_{X_{k-1}}' A^T (I_n - (V_N + A\hat{V}_{X_{k-1}} A^T)^{-1} A\hat{V}_{X_{k-1}}') V_N^{-1}
\] (C.11)

\[
= \hat{V}_{X_{k-1}}' A^T (V_N + A\hat{V}_{X_{k-1}} A^T)^{-1},
\] (C.12)

where the last step follows from factorizing the middle term in (C.11) by \((V_N + A\hat{V}_{X_{k-1}} A^T)^{-1}\) from the left.

Combining (C.8) and (C.12) in the upper right corner formula in (A.6), we finally have

\[
V_{X_{k-1}, X_k} = V_{X'_{k-1}, X_k}
\] (C.13)

\[
= “- A^{-1}B(D - CA^{-1}B)^{-1}”
\] (C.14)

\[
= \hat{V}_{X'_{k-1}} A^T \tilde{W}_{X_k} \hat{V}_{X_k}
\] (C.15)

\[
= F_{k-1} \hat{V}_{X_{k-1}} A^T (I_n - \tilde{W}_{X_k} \hat{V}_{X_k}).
\] (C.16)


C.2 Cross-Covariance Between $X_{k-1}$ and $U_k$

Integrating (C.1) over $x_k$, we obtain

$$p(x'_{k-1}, u_k | y, \theta) \propto \mathcal{N}(x'_{k-1} : \hat{m}_{X_{k-1}'}, \hat{V}_{X_{k-1}'}) \mathcal{N}(u_k : 0, \Sigma_{U_k}) \cdot \mathcal{N}(\hat{m}_{X_k}: Ax'_{k-1} + B_k u_k, V_E + \hat{V}_{X_k}). \quad \text{(C.17)}$$

Thus, denoting

$$V_M = V_E + \hat{V}_{X_k}, \quad \text{(C.18)}$$

the precision matrix of the joint density (C.17) is

$$W_{(X'_{k-1}, U_k)} = \begin{bmatrix} \hat{V}_{X'_{k-1}}^{-1} + A^T V_M^{-1} A & A^T V_M^{-1} B_k \\ B_k^T V_M^{-1} A & \Sigma_{U_k}^{-1} + B_k^T V_M^{-1} B_k \end{bmatrix}. \quad \text{(C.19)}$$

We now use the block-wise inversion formula (A.6) to compute the cross-covariance matrix. Referring to the notation in (A.6), we get

“$(D - CA^{-1}B)$”

$$= \Sigma_{U_k}^{-1} + B_k^T V_M^{-1} B_k - B_k^T V_M^{-1} A(\hat{V}_{X'_{k-1}}^{-1} + A^T V_M^{-1} A)^{-1} A^T V_M^{-1} B_k$$

$$= \Sigma_{U_k}^{-1} + B_k^T (V_M + A\hat{V}_{X'_{k-1}} A^T)^{-1} B_k,$$

and thus

“$(D - CA^{-1}B)^{-1}$”

$$= (\Sigma_{U_k}^{-1} + B_k^T (V_M + A\hat{V}_{X'_{k-1}} A^T)^{-1} B_k)^{-1} \quad \text{(C.20)}$$
$$= \Sigma_{U_k} - \Sigma_{U_k} B_k^T (V_M + A\hat{V}_{X'_{k-1}} A^T + B_k \Sigma_{U_k} B_k^T)^{-1} B_k \Sigma_{U_k} \quad \text{(C.21)}$$
$$= \Sigma_{U_k} - \Sigma_{U_k} B_k^T \hat{W}_{X_k} B_k \Sigma_{U_k}. \quad \text{(C.22)}$$

Furthermore,

$$B_k \left( D - CA^{-1}B \right)^{-1} = B_k(\Sigma_{U_k} - \Sigma_{U_k} B_k^T \hat{W}_{X_k} B_k \Sigma_{U_k}) \quad \text{(C.23)}$$
$$= (\hat{W}_{X_k}^{-1} - B_k \Sigma_{U_k} B_k^T) \hat{W}_{X_k} B_k \Sigma_{U_k} \quad \text{(C.24)}$$
$$= (V_M + A\hat{V}_{X'_{k-1}} A^T) \hat{W}_{X_k} B_k \Sigma_{U_k}. \quad \text{(C.25)}$$

On the other hand, we also have

“$-A^{-1}B$”

$$= -(\hat{V}_{X'_{k-1}}^{-1} + A^T V_M^{-1} A)^{-1} A^T V_M^{-1} B_k \quad \text{(C.26)}$$
$$= -\hat{V}_{X'_{k-1}} A^T (V_M + A\hat{V}_{X'_{k-1}} A^T)^{-1} B_k, \quad \text{(C.27)}$$
using similar steps as in (C.9)–(C.12).

Combining (C.25) with (C.27), we thus obtain

\[
V_{X_{k-1},U_k} = V_{X_{k-1}',U_k} = -A^{-1} B \left( D - CA^{-1} B \right)^{-1} \tag{C.28}
\]

\[
= -\tilde{V}_{X_{k-1}'} A^T \tilde{W}_{X_k} B_k \Sigma_U \tag{C.29}
\]

\[
= -F_{k-1} \tilde{V}_{X_{k-1}} A^T \tilde{W}_{X_k} B_k \Sigma_U. \tag{C.30}
\]

\[
V_{X_{k-1},U_k} = \left( I_n - \tilde{V}_{X_k} \tilde{W}_{X_k} \right) B_k \Sigma_U, \tag{C.31}
\]

C.3 Cross-Covariance Between \( X_k \) and \( U_k \)

Integrating (C.1) over \( x_k \), we obtain

\[
p(u_k, x_k | y, \theta) \propto N(u_k : 0, \Sigma_{U_k}) N(x_k : \tilde{m}_{X_k}, \tilde{V}_{X_k}) \cdot N(x_k : A \tilde{m}_{X_{k-1}} + B_k u_k, A \tilde{V}_{X_{k-1}'} \tilde{A}^T + V_E). \tag{C.32}
\]

Thus, denoting

\[
V_T = A \tilde{V}_{X_{k-1}'} \tilde{A}^T + V_E, \tag{C.33}
\]

the precision matrix of the joint density (C.32) is

\[
W_{(X_k,U_k)} = \begin{bmatrix}
V_T^{-1} + \tilde{V}_{X_k}^{-1} & -V_T^{-1} B_k \\
-B_k^T V_T^{-1} & \Sigma_{U_k}^{-1} + B_k^T V_T^{-1} B_k
\end{bmatrix}. \tag{C.34}
\]

Looking at the similarities between (C.34) and (C.19)

\[
V_T \leftrightarrow V_M \tag{C.35}
\]

\[
\tilde{V}_{X_k} \leftrightarrow \tilde{V}_{X_{k-1}'} \tag{C.36}
\]

\[
-I_n \leftrightarrow A, \tag{C.37}
\]

we can deduce that

\[
V_{X_k,U_k} = \tilde{V}_{X_k} \tilde{W}_{X_k} B_k \Sigma_U \tag{C.38}
\]

\[
= \left( I_n - \tilde{V}_{X_k} \tilde{W}_{X_k} \right) B_k \Sigma_U, \tag{C.39}
\]

by adapting (C.30) accordingly.
Appendix D

General Quadratic Form Minimization

Given a set \( \mathcal{A} \subset \mathbb{R}^{n \times n} \) and a general quadratic cost function in \( A \in \mathbb{R}^{n \times n} \)
\[
F(A) = \text{tr} \left( (AW_A A^T - 2A_\xi A)W_E \right),
\]
with \( W_E \) and \( W_A \) symmetric positive definite matrices of \( \mathbb{R}^{n \times n} \) and \( \xi_A \in \mathbb{R}^{n \times n} \), we wish to find
\[
\hat{A} = \arg \min_{A \in \mathcal{A}} F(A)
\]
\[
= \arg \min_{A \in \mathcal{A}} \text{tr} \left( (AW_A A^T - 2A_\xi A)W_E \right).
\]

This optimization problem appears several times in this thesis and in particular when estimating the state transition matrix of a LSSM.

We here provide closed-form solutions for \( \hat{A} \) for different and popular choices of \( \mathcal{A} \).

D.1 General Structured Matrix

Suppose that \( A \) is structured such that some of its coefficients are constant and the rest are unknown reals. In other words,
\[
A = P_0 + \sum_{q=1}^{Q} a_q P_q,
\]

211
where $P_0 \in \mathbb{R}^{n \times n}$ contains the constant coefficients of $A$, $a_q \in \mathbb{R}$ are the coefficient of $A$ to be estimated and $P_q \in \{0,1\}^{n \times n}$ is zero everywhere except at the positions of $a_q$ in the matrix $A$.

Then, $F(A)$ writes as

$$F(A) = a^T W_a a - 2a^T \xi_a + \text{tr} \left( (P_0 W_A P_0^T - 2P_0 \xi_A) W_E \right),$$

with $a = [a_1 \ldots a_Q]^T \in \mathbb{R}^Q$, $W_a \in \mathbb{R}^{Q \times Q}$, and $\xi_a \in \mathbb{R}^Q$ such that their coefficients are

$$\{W_a\}_{i,j} = \text{tr} \left( P_i W_A P_j^T W_E \right) \quad \text{(D.6)}$$

$$\{\xi_a\}_i = \text{tr} \left( P_i \xi_A W_E \right) - \text{tr} \left( P_i W_A P_0^T W_E \right), \quad \text{(D.7)}$$

for all $(i,j) \in \{1,\ldots,Q\}^2$. It follows that $W_a$ is symmetric positive (semi-)definite. Finally, (D.3) simplifies into

$$\hat{a} = \arg\min_{a \in \mathbb{R}^Q} a^T W_a a - 2a^T \xi_a$$

$$= W_a^{-1} \xi_a, \quad \text{(D.9)}$$

from which we can deduce $\hat{A}$ using (D.4). The equivalence between (D.8) and (D.9) holds only when $W_a$ is positive definite. If $W_a$ is not invertible, $\hat{a}$ is not unique but replacing $W_a^{-1}$ by the pseudo-inverse of $W_a$ in (D.9) provides a solution.

Despite the appearing computational complexity of (D.6) and (D.7), many elements of the matrices $P_i$, $i \in \{1,\ldots,Q\}$ are often zero and thus, the trace expressions involve only a few computations.

This “General Structured Matrix” optimization also holds for non-square matrices provided that (D.3) makes sense.

### D.2 Controllable Canonical Form

If $A$ is in controllable canonical form, then

$$A = \begin{bmatrix} 0 & 1 & & \cdot & \cdot & \cdot \\ & \ddots & \ddots & \ddots & & \\ & & 0 & 1 \\ a_n & \ldots & a_2 & a_1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

(D.10)

can be decomposed as in (D.4) where $P_0$ has ones in the first upper diagonal and zeros elsewhere, $Q = n$, and $P_q$ has zeros everywhere except for the $(n-q+1)^{th}$ coefficient of the last row that is 1. Thus, $\hat{A}$ is found using Sec. D.1.
D.3 Observable Canonical Form

If $A$ is in observable canonical form, then

$$A = \begin{bmatrix} 0 & a_n \\ 1 & \ddots & \iddots \\ \ddots & 0 & a_2 \\ 1 & & a_1 \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{D.11}$$

can be decomposed as in (D.4) where $P_0$ has ones in the first lower diagonal and zeros elsewhere, $Q = n$, and $P_q$ has zeros everywhere except for the $(n - q + 1)^{th}$ coefficient of the last column that is 1. Thus, $\hat{A}$ is found using Sec. D.1.

D.4 Jordan Canonical Form

For $m \in \mathbb{N}$, let $N_m \in \mathbb{R}^{m \times m}$ denotes the nilpotent matrix having ones in the first upper diagonal and zeros elsewhere.

If $A$ is in block Jordan form, then

$$A = \text{diag}(J_1, \ldots, J_Q, G_1, \ldots, G_L), \tag{D.12}$$

where $J_q$ denotes a Jordan block corresponding to a real eigenvalue $\lambda_q \in \mathbb{R}$ of multiplicity $\pi_q \in \mathbb{N}$, i.e.,

$$J_q = \lambda_q I_{\pi_q} + N_{\pi_q} \in \mathbb{R}^{\pi_q \times \pi_q}, \tag{D.13}$$

and $G_\ell$ denotes a Jordan block corresponding to a pair of complex conjugate eigenvalues $\alpha_\ell \pm i\beta_\ell$, $(\alpha_\ell, \beta_\ell) \in \mathbb{R} \times \mathbb{R}^*$, of multiplicity $\pi_\ell \in \mathbb{N}$, i.e.,

$$G_\ell = I_{\pi_\ell} \otimes (\alpha_\ell I_2 + \beta_\ell S_2) + N_{\pi_\ell} \otimes I_2 \in \mathbb{R}^{2\pi_\ell \times 2\pi_\ell}, \tag{D.14}$$

where $\otimes$ denotes the Kronecker product and

$$S_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{D.15}$$

Note that we have

$$\alpha_\ell I_2 + \beta_\ell S_2 = \begin{bmatrix} \alpha_\ell & -\beta_\ell \\ \beta_\ell & \alpha_\ell \end{bmatrix}. \tag{D.16}$$
Combining (D.13) and (D.14) with (D.12), $A$ can be written as

$$
A = P_0 + \sum_{q=1}^{Q} \lambda_q P_q + \sum_{\ell=1}^{L} \left( \alpha_\ell P^{(1)}_\ell + \beta_\ell P^{(2)}_\ell \right), \quad (D.17)
$$

where for $q \in \{1, \ldots, Q\}$, $\ell \in \{1, \ldots, L\},$

$$
\begin{align*}
P_q &= T(q, I_{\pi_q}) \quad (D.18) \\
P^{(1)}_\ell &= \tilde{T}(\ell, I_{\pi_\ell} \otimes I_2) \quad (D.19) \\
P^{(2)}_\ell &= \tilde{T}(\ell, I_{\pi_\ell} \otimes S_2) \quad (D.20) \\
P_0 &= \sum_{q=1}^{Q} T(q, N_{\pi_q}) + \sum_{\ell=1}^{L} \tilde{T}(\ell, N_{\pi_\ell} \otimes I_2), \quad (D.21)
\end{align*}
$$

with $T(q, W)$ being a block diagonal matrix as in (D.12) with all blocks being zero except for block $J_q$ that is replaced by $W \in \mathbb{R}^{\pi_q \times \pi_q}$ and with $\tilde{T}(\ell, W)$ being a block diagonal matrix as in (D.12) with all blocks being zero except for block $G_\ell$ that is replaced by $W \in \mathbb{R}^{2\pi_\ell \times 2\pi_\ell}$.

Since (D.17) is of the form (D.4), the parameters $\hat{\lambda}_q \in \mathbb{R}$, $q \in \{1, \ldots, Q\}$ and $(\hat{\alpha}_\ell, \hat{\beta}_\ell) \in \mathbb{R} \times \mathbb{R}^*$, $\ell \in \{1, \ldots, L\}$, and thus $\hat{A}$ are found using Sec. D.1.

In case $W_E$ is a block diagonal matrix of similar block sizes as in (D.12) (e.g., $W_E = I_n$), the optimization in (D.3) splits for each Jordan block $J$ ($J$ being either $J_q$ or $G_\ell$) since

$$
F(A) = \sum_{\text{blocks } J} \text{tr} \left( (JW_A^{(J)} J^T - 2J\xi_A^{(J)})W_E^{(J)} \right), \quad (D.22)
$$

where $W_A^{(J)}$ is the diagonal block of $W_A$ corresponding to block $J$ (i.e., when decomposing $W_A$ in the same block structure as $A$), $\xi_A^{(J)}$ is the part of $\xi_A$ that is effectively multiplied by block $J$, and $W_E^{(J)}$ is the diagonal block of $W_E$ corresponding to block $J$. Then, each optimization over a block $J$

$$
\hat{J} = \arg\min_{J} \text{tr} \left( (JW_A^{(J)} J^T - 2J\xi_A^{(J)})W_E^{(J)} \right), \quad (D.23)
$$

is done as described before, which simplifies the formulas. More specifically, if $J$ is a Jordan block of the first form

$$
J = \lambda I_{\pi} + N_{\pi} \in \mathbb{R}^{\pi \times \pi}, \quad (D.24)
$$
the optimum $\lambda$ is given by
\[
\hat{\lambda} = \frac{\text{tr} \left( \xi_A^{(J)} W_E^{(J)} \right) - \text{tr} \left( W_A^{(J)} N_{2\pi} W_E^{(J)} \right)}{\text{tr} \left( W_A^{(J)} W_E^{(J)} \right)} .
\] (D.25)

On the other hand, if $J \in \mathbb{R}^{2\pi \times 2\pi}$ is a Jordan block of the second form
\[
J = I_{\pi} \otimes (\alpha I_2 + \beta S_2) + N_{\pi} \otimes I_2
\] (D.26)
with $S_2 = I_{\pi} \otimes S_2$, the optima are given by
\[
\begin{bmatrix}
\text{tr} \left( W_A^{(J)} W_E^{(J)} \right) \\
\text{tr} \left( S_{2\pi} W_A^{(J)} W_E^{(J)} \right)
\end{bmatrix}
= \begin{bmatrix}
\hat{\alpha} \\
\hat{\beta}
\end{bmatrix}
\] (D.28)

D.5 Modified Jordan Canonical Form

When estimating $A$ in Jordan canonical form, we observe that when a block degenerates, we have
\[
\lim_{\lambda_q \to 0} J_q = N_{\pi q}
\] (D.29)
\[
\lim_{(\alpha_\ell, \beta_\ell) \to 0} G_\ell = N_{\pi \ell} \otimes I_2,
\] (D.30)
which still creates a finite impulse response system. Instead, we prefer to use the modified Jordan canonical form where
\[
J_q = \lambda_q (I_{\pi q} + N_{\pi q})
\] (D.31)
\[
G_\ell = (I_{\pi \ell} + N_{\pi \ell}) \otimes (\alpha_\ell I_2 + \beta_\ell S_2),
\] (D.32)
instead of (D.13) and (D.14). When such blocks degenerate, they become zero and do not affect the measurements anymore. Furthermore, (D.17) still holds with the quantities
\[
P_q = T(q, I_{\pi q} + N_{\pi q})
\] (D.33)
\[
P^{(1)}_\ell = \bar{T}(\ell, (I_{\pi \ell} + N_{\pi \ell}) \otimes I_2)
\] (D.34)
\[
P^{(2)}_\ell = \bar{T}(\ell, (I_{\pi \ell} + N_{\pi \ell}) \otimes S_2)
\] (D.35)
\[
0 = 0,
\] (D.36)
and thus \( \hat{A} \) is also found using Sec. D.1.

This parametrization forbids the use of finite impulse response systems, which thus need to be included explicitly if required. For instance, we can introduce a first block \( J_0 = N_{\pi_0} \), which is constant and can be included in \( P_0 \).
Appendix E

Covariance Estimation With NUV Regularization

E.1 Independent and Identically Distributed Samples

We wish to estimate $H \in \mathbb{R}^{M \times J}$ from independent observations $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$, given by

$$y_k = H u_k + Z_k,$$

(E.1)

with $u_k \in \mathbb{R}^J$ and $Z_k \sim \mathcal{N}(0, \sigma_Z^2 I_M)$. Thus, we have

$$p(y|u, H, \sigma_Z^2) = \prod_{k=1}^{K} \mathcal{N}(y_k : H u_k, \sigma_Z^2 I_M).$$

(E.2)

We further assume that

$$U_k \sim \mathcal{N}(0, \Sigma),$$

(E.3)
with unknown diagonal covariance matrix $\Sigma = \text{diag}(\sigma^2)$ with $\sigma^2 = (\sigma_1^2, \ldots, \sigma_J^2)$. Consequently, the likelihood is

$$ p(y|\sigma^2, H, \sigma_Z^2) = \int p(y|u, H, \sigma_Z^2)p(u|\sigma^2) \, du $$

$$ = \prod_{k=1}^{K} \int \mathcal{N}(y_k : Hu_k, \sigma_Z^2 I_M) \mathcal{N}(u_k : 0, \Sigma) \, du_k $$

$$ = \prod_{k=1}^{K} \mathcal{N}(y_k : 0, \sigma_Z^2 I_M + H\Sigma H^T). $$

Thus, maximizing the likelihood is equivalent to minimizing the cost

$$ J(\sigma^2, H, \sigma_Z^2) = \text{tr} \left( (\sigma_Z^2 I_M + H\Sigma H^T)^{-1} S_K \right) + \ln |\sigma_Z^2 I_M + H\Sigma H^T|, $$

where $S_K \in \mathbb{R}^{M \times M}$ denotes the sample covariance matrix

$$ S_K = \frac{1}{K} \sum_{k=1}^{K} y_k y_k^T. $$

The function (E.7) only depends on the matrix

$$ V_{\text{tot}} = \sigma_Z^2 I_M + H\Sigma H^T \in \mathbb{R}^{M \times M}, $$

and is actually convex in $V_{\text{tot}}^{-1}$. The matrix $V_{\text{tot}}$ can be interpreted as the regularized covariance matrix to be estimated.

When $J \geq M$, $H\Sigma H^T$ is any symmetric positive semi-definite matrix. When $J < M$, $H\Sigma H^T$ is any symmetric positive semi-definite matrix of rank at most $J$. In any case, any symmetric positive semi-definite matrix can be written as

$$ H\Sigma H^T = VDV^T, $$

with $V \in \mathcal{O}_M(\mathbb{R})$ an orthogonal matrix and $D = \text{diag}(d)$, with $d = (d_1, d_2, \ldots, d_M)$ in decreasing order (i.e., $d_i \geq d_{i+1}$). Thus, we obtain

$$ J(\sigma^2, H, \sigma_Z^2) = \text{tr} \left( (\sigma_Z^2 I_M + VDV^T)^{-1} S_K \right) + \ln |\sigma_Z^2 I_M + VDV^T| $$

$$ = \text{tr} \left( (\sigma_Z^2 I_M + D)^{-1} V^T S_K V \right) + \ln |\sigma_Z^2 I_M + D| $$

$$ = \text{tr} \left( (\sigma_Z^2 I_M + D)^{-1} V^T U \Lambda U^T V \right) + \sum_{i=1}^{M} \ln(\sigma_Z^2 + d_i) $$

$$ = J(d, V, \sigma_Z^2), $$
where we have used an eigenvalue decomposition

\[ S_K = U \Lambda U^T, \tag{E.15} \]

with an orthogonal matrix \( U \in \mathcal{O}(\mathbb{R}) \) and \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_M), \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \geq 0. \)

Using the trace inequality (A.11), we have

\[ \text{tr} \left( (\sigma_Z^2 I_M + D)^{-1} V^T U \Lambda U^T V \right) \geq \text{tr} \left( (\sigma_Z^2 I_M + D)^{-1} \Lambda \right), \tag{E.16} \]

for all orthogonal matrices \( V \) (since \( U^T V \) is orthogonal) and equality holds for \( U^T V = I_M \). Thus, for any diagonal matrix \( D \) and non negative real \( \sigma_Z^2 \),

\[ \hat{V} = \arg\min_{V \in \mathcal{O}} J(d, V, \sigma_Z^2) = U. \tag{E.17} \]

It follows that

\[ \min_{V \in \mathcal{O}} J(d, V, \sigma_Z^2) = \sum_{i=1}^M \frac{\lambda_i}{\sigma_Z^2 + d_i} + \ln(\sigma_Z^2 + d_i). \tag{E.18} \]

For a fixed \( \sigma_Z^2 \), we have

\[ \hat{d}_i = \max(0, \lambda_i - \sigma_Z^2). \tag{E.19} \]

Thus, the number of non zero coefficients of \( \hat{d} \) is the number of eigenvalues of \( S_K \) that are strictly bigger than \( \sigma_Z^2 \). However, if \( J < M \), we force the matrix \( H \Sigma H^T = V D V^T \) to have a rank of at most \( J \), i.e., we have the extra constraint that the number of non-zero coefficients \( d_i \) is at most \( J \). If \( \hat{d} \) does not satisfy this property, we only keep the first (i.e., the largest) \( J \) components of \( \hat{d} \) and set the remaining ones to zero. Finally, all matrices \( H \in \mathbb{R}^{M \times J} \) and \( \Sigma = \text{diag}(\sigma^2) \in \mathbb{R}^{J \times J} \) satisfying

\[ H \Sigma H^T = \hat{V} \hat{D} \hat{V}^T, \tag{E.20} \]

minimizes the cost function \( J(\sigma^2, H, \sigma_Z^2) \) for a fixed \( \sigma_Z^2 \).

In case \( \sigma_Z^2 \) is unknown and estimated by maximum likelihood we have the following cases.

- \( J \geq M \): choosing \( \sigma_Z^2 \) such that for all \( i \in \{1, \ldots, M\} \),

\[ \lambda_i - \sigma_Z^2 \geq 0, \tag{E.21} \]
is optimum since the value of each summand in (E.18) cannot be decreased further. Thus, any $\sigma_Z^2$ satisfying $0 \leq \sigma_Z^2 \leq \lambda_{\min}$ is a valid estimate. Note that in this situation we have $\hat{D} = \Lambda - \sigma_Z^2 I_M$ and thus, for any optimum $(\hat{\sigma}^2, \hat{H}, \hat{\sigma}_Z^2)$, we have

$$\hat{V}_{\text{tot}} = S_K.$$  

(E.22)

This result is not a surprise since

$$\{\sigma_Z^2 I_M + H\Sigma H^T : \sigma_Z^2 \in \mathbb{R}_+, H \in \mathbb{R}^{M \times J}, \sigma^2 \in \mathbb{R}_+^J\}, = \mathbb{S}_+$$  

(E.23)

when $J \geq M$, which thus reduces our problem to the non-regularized covariance estimation problem.

• $J < M$: since anyway $\hat{d}_i = 0$ for $i > J$, we need to choose $\sigma_Z^2 \leq \lambda_J$ such that

$$\lambda_i - \sigma_Z^2 \geq 0,$$  

(E.24)

is satisfied at least for $i \in \{1, \ldots, J\}$. With this condition, we have

$$\min_d \min_{V \in \mathcal{O}} J(d, V, \sigma_Z^2) = \sum_{i=1}^J 1 + \ln(\lambda_i) + \sum_{i=J+1}^M \frac{\lambda_i}{\sigma_Z^2} + \ln \sigma_Z^2. \quad \text{ (E.25)}$$

It follows that

$$\hat{\sigma}_Z^2 = \frac{1}{M - J} \sum_{i=J+1}^M \lambda_i,$$  

(E.26)

which indeed satisfies $\hat{\sigma}_Z^2 \leq \lambda_J$.

This problem is known as the Probabilistic PCA problem [70] and the sensible PCA (SPCA) problem [64]. In the limit case where $\sigma_Z^2 \to 0$, this is a PCA problem.

Note that the maximum likelihood solution requires to compute the eigenvalue decomposition of the sample covariance matrix. This can be avoided using, for instance, an EM algorithm.

Even if the case $J < M$ is interesting, the case where $J \geq M$ is the one we typically encounter in this thesis. Often, we further want to obtain an estimated diagonal matrix $\Sigma$ with a lot of zeros in the diagonal such that

$$H\Sigma H^T = \sum_{j=1}^J \sigma_j^2 h_j h_j^T,$$  

(E.27)
is a sum of only few rank one matrices ($h_j$ denote the $j$th column of $H$). Considering $\sigma^2_Z$ fixed, the current regularization does not directly promote sparse variances $\sigma^2$ but rather constrains the rank of the matrix $H\Sigma H^\top$. Nevertheless, among all solutions $(\sigma^2, H)$ satisfying (E.20) some of them actually consist of a sparse estimated $\sigma^2$. The sparsest solution is characterized by a number of non zero elements equal to the rank of $H\Sigma H^\top$, which is prescribed by a choice of $\sigma^2_Z$.

To enforce sparse solution, constraining $H$ to a given set $H \subset \mathbb{R}^{M \times J}$ can be beneficial. Indeed, assuming that

$$\hat{V} \hat{D} \hat{V}^\top \in \{H\Sigma H^\top : H \in H, \sigma^2 \in \mathbb{R}^J_+\},$$

(E.28)

the number of solutions $(\sigma^2, H)$ satisfying (E.20) reduces and might favor sparse decompositions. For instance, excluding equal columns (up to a scale factor) in $H$ already disregards a lot of non-really sparse solutions satisfying (E.20).

Note that estimating $\sigma^2_Z$ is usually not a good idea since the regularization we have introduced essentially disappears unless we have suitable constraints on $H$.

### E.2 The Single Sample Case

We wish to estimate $H \in \mathbb{R}^{M \times J}$ from a single observation vector $y \in \mathbb{R}^M$, given by

$$y = HU + Z,$$

(E.29)

with $U \in \mathbb{R}^J$ and $Z \in \mathbb{R}^M$ mutually independent Gaussian random vectors such that

$$Z \sim \mathcal{N}(0, \sigma^2_Z I_M),$$

$$U \sim \mathcal{N}(0, \Sigma),$$

(E.30)  
(E.31)

with unknown diagonal covariance matrix $\Sigma = \text{diag}(\sigma^2)$ with $\sigma^2 = (\sigma_1^2, \ldots, \sigma_J^2)$. We estimate all unknown parameters by maximum likelihood.

This is a special case of the problem described in Sec. E.1 where the number of samples is one (i.e., $K = 1$). In this case, the sample covariance matrix $S_1$ has a trivial eigenvalue decomposition

$$S_1 = yy^\top = U \text{diag}(\lambda) U^\top,$$

(E.32)
with $\lambda_1 = \|y\|^2$, $\lambda_m = 0$ for $m > 1$, and $U$ such that the first column is $\frac{y}{\|y\|^2}$ and the remaining columns such that $U$ is an orthogonal matrix. Assuming that $\|y\|^2 > \sigma_Z^2$, it follows that all matrices $H$ and diagonal matrices $\Sigma$ satisfying

$$H\Sigma H^T = \frac{\|y\|^2 - \sigma_Z^2}{\|y\|^2}yy^T$$

(E.33)

maximizes the likelihood function. Thus, in any case, we obtain

$$\hat{V}_{\text{tot}} = \sigma_Z^2 I_M + \frac{\|y\|^2 - \sigma_Z^2}{\|y\|^2}yy^T.$$  

(E.34)

Furthermore, plugging this optimum value in the cost function (E.7), we have

$$\min_{H, \sigma^2} J(\sigma^2, H, \sigma_Z^2) = 1 + (M - 1) \ln \sigma_Z^2 + \ln \|y\|^2,$$

(E.35)

after computations that involve (A.4) and (A.5). Minimizing (E.35) with respect to $\sigma_Z^2$ leads to $\hat{\sigma}_Z^2 = 0$, which is usually undesirable and provides the sample covariance matrix as estimated covariance matrix, i.e.,

$$\hat{V}_{\text{tot}} = yy^T.$$  

(E.36)

In practical cases, $H$ needs to be highly regularize by constraining $H$ to a given set $\mathbb{H} \subset \mathbb{R}^{M \times J}$. Then, for a fixed $\sigma_Z^2$, (E.35) provides a lower bound for the minimum cost

$$\min_{H \in \mathbb{H}, \sigma^2} J(\sigma^2, H, \sigma_Z^2) \geq 1 + (M - 1) \ln \sigma_Z^2 + \ln \|y\|^2.$$  

(E.37)

### E.3 A Dictionary Learning Problem With Independent Samples

We wish to estimate $H \in \mathbb{R}^{M \times J}$ from independent observations $y_k \in \mathbb{R}^M$, $k \in \{1, \ldots, K\}$, given by

$$y_k = Hu_k + Z_k,$$  

(E.38)

with $u_k \in \mathbb{R}^J$ and $Z_k \iid \mathcal{N}(0, \sigma_Z^2 I_M)$. Furthermore, we assume that the $u_k$’s are independent with

$$U_k \sim \mathcal{N}(0, \Sigma_k),$$  

(E.39)
where $\Sigma_k$ is an unknown diagonal matrix, estimated by maximum likelihood. Unlike Sec. E.1, $\Sigma_k$ depends on the sample index $k$.

The maximum likelihood estimate of $(\Sigma_k, \ldots, \Sigma_K, H)$ minimizes

$$
\sum_{k=1}^{K} y_k^T (\sigma_Z^2 I_M + H\Sigma_k H^T)^{-1} y_k + \ln |\sigma_Z^2 I_M + H\Sigma_k H^T|, \quad (E.40)
$$

which corresponds to the negative log-likelihood up to irrelevant constants.

There is no known closed-form solution. However, an EM algorithm is available by considering $U_k$ as hidden variable. The EM step consists in maximizing

$$
\mathbb{E}[p(y_1, \ldots, y_K, U_1, \ldots, U_K | H, \Sigma_1, \ldots, \Sigma_K)] = \sum_{k=1}^{K} \mathbb{E}[\ln \mathcal{N}(y_k | H u_k, \sigma_Z^2 I_M)] + \sum_{k=1}^{K} \mathbb{E}[\ln \mathcal{N}(u_k | 0, \Sigma_k)]. \quad (E.41)
$$

Thus, we obtain the EM update

$$
\{\hat{\Sigma}_k\}_{j,j} = \mathbb{E}\left[\{U_k\}_j^2\right], \quad j \in \{1, \ldots, J\} \quad (E.42)
$$

$$
\hat{H} = \left(\sum_{k=1}^{K} y_k \mathbb{E}[U_k]^T\right) \left(\sum_{k=1}^{K} \mathbb{E}[U_k U_k^T]\right)^{-1}. \quad (E.43)
$$
Bibliography


About the Author

Nour Zalmaï was born in Gray, France in 1989. He attended high school at Lycée Cournot, Gray, France and obtained the French scientific Baccalauréat with highest honour in 2007.

From 2007 to 2009, he pursued his education with intensive courses in Mathematics and Physics (“classe préparatoire aux grandes écoles”) at Lycée du Parc in Lyon to prepare for the nationwide competitive exams.

In 2009, he joined Supélec (“École Supérieure d’Électricité”) in Gif-sur-Yvette, France, which is a leading school in engineering. Throughout his studies at Supélec, he was ranked amongst the top students.

During the summer 2011, he visited the Unit of Neuroscience, Information and Complexity (UNIC) at CNRS in Gif-sur-Yvette, as a research engineer.

In 2011, he decided to attend the master program in electrical engineering at ETH Zurich, within a double degree program and partly supported by an Erasmus scholarship. Thus, in 2013, he obtained both the “Diplôme d’Ingénieur Supélec” and the M.Sc. in Information Technology and Electrical Engineering from ETH Zurich.

Since 2013, he has been a PhD candidate and a full research assistant at the Signal and Information Processing Laboratory (ISI) at ETH Zurich. His focus is on graphical models for learning sparse signal decompositions.
Series in Signal and Information Processing

edited by Hans-Andrea Loeliger


Vol. 18: Volker Maximillian Koch, A Factor Graph Approach to Model-Based Signal Separation. ISBN 3-86628-140-4


Vol. 20: Maja Ostojc, Multitree Search Decoding of Linear Codes. ISBN 3-86628-363-6


Hartung-Gorre Verlag Konstanz – http://www.hartung-gorre.de