Parametric Time–Frequency Analysis for Discrimination of Non–Stationary Signals

Luis David Avendaño Valencia

A Thesis presented for the degree of Master of Science

Control and Digital Signal Processing Group
Department of Electric and Electronic Engineering
Universidad Nacional de Colombia – Manizales
Colombia
May 2009
Contents

1 Introduction ................................................................. 1
  1.1 Problem statement .................................................. 2
  1.2 Objectives ............................................................. 4
    1.2.1 General objective ............................................ 4
    1.2.2 Specific objectives .......................................... 4

2 Random signals and systems ............................................. 5
  2.1 Intro ................................................................. 5
  2.2 Summary of probability theory .................................. 5
  2.3 Random discrete–time signals .................................. 7
  2.4 Stationary Time Series ........................................... 8
  2.5 Power spectrum ..................................................... 9
  2.6 Discrete time systems with random inputs ..................... 9
    2.6.1 Autocorrelation function of the output .................. 10
    2.6.2 Power spectral density of the output .................... 11
  2.7 Linear random process models ................................ 11
  2.8 Power Spectral Density of ARMA processes .................... 13
  2.9 Spectral Decomposition ........................................... 14

3 Estimation theory ........................................................... 16
  3.1 Estimation Problem ................................................ 16
  3.2 Bayesian estimation ................................................ 18
    3.2.1 Mean square estimation .................................... 19
    3.2.2 Maximum a posteriori estimation ......................... 20
    3.2.3 Maximum likelihood estimation ........................... 21
  3.3 Least squares estimation .......................................... 21
    3.3.1 Empirical risk minimization principle .................... 22
    3.3.2 Least squares estimation from experimental data ....... 23
  3.4 Recursive estimation ................................................. 23
    3.4.1 Least mean square algorithm ............................. 24
## Contents

3.4.2 Recursive Least Squares Algorithm ........................................... 24
3.4.3 Recursive Bayesian Estimation .................................................. 25
3.4.4 Kalman filter ............................................................................ 27

4 Non–stationary models ........................................................................ 32
4.1 Introduction .................................................................................. 32
4.2 Non-parametric approach ............................................................... 33
4.3 Parametric approach ...................................................................... 35
4.4 Parameter estimation of TVAR models ........................................... 36
  4.4.1 Locally stationary methods ....................................................... 37
  4.4.2 Adaptive methods .................................................................... 37
  4.4.3 Smoothness priors method ....................................................... 39
  4.4.4 Functional approximation methods ......................................... 40
4.5 Estimation of time–dependent variance ........................................... 42
4.6 Model order estimation ................................................................... 43
4.7 Properties of TVAR models ........................................................... 44
  4.7.1 Impulse response function ....................................................... 45
  4.7.2 Covariance function of a TVAR model ..................................... 46
4.8 Time–frequency distributions ......................................................... 46

5 Dimensionality reduction in TFR ......................................................... 49
5.1 Feature extraction by tiling of matrices ............................................ 51
5.2 Feature extraction with linear decomposition methods ...................... 53
  5.2.1 Principal component analysis .................................................... 53
  5.2.2 Partial Least Squares ............................................................... 57

6 Classification of phonocardiographic signals ..................................... 58
6.1 Database ....................................................................................... 60
6.2 Preprocessing ................................................................................ 60
6.3 TVAR model set up ........................................................................ 61
  6.3.1 Model order estimation ............................................................ 61
  6.3.2 Estimation of TVAR parameters ............................................. 62
  6.3.3 Estimation of parametric TFR .................................................. 65
6.4 Feature extraction from TFR ........................................................... 66
  6.4.1 Working area selection ............................................................ 66
  6.4.2 Tiling ...................................................................................... 68
  6.4.3 Linear decomposition methods ............................................... 68
6.5 Feature extraction from TVAR parameters ..................................... 71
6.6 Classification and validation ........................................................... 72
  6.6.1 Number of neighbors tuning ..................................................... 73
6.6.2 Robustness of feature extraction methods 74
6.6.3 Robustness of feature estimation methods 76
6.6.4 Classification with TVAR parameters 77
6.6.5 Feature selection by relevance analysis 78

6.7 Summary of results of tuned methodologies 79
6.8 Discussion and Conclusions 81

7 Detection of epilepsy 84
7.1 Database 87
7.2 TVAR model set up 88
  7.2.1 Model order estimation 88
  7.2.2 Estimation of TVAR parameters and parametric TFR 89
7.3 Feature extraction from TFR 89
  7.3.1 Selection of working area by variability and relevance analysis 89
  7.3.2 Feature extraction with tiling 91
  7.3.3 Feature extraction with linear decomposition methods 92
7.4 Summary of results of tuned methodologies 94
  7.4.1 Classification problem 1 94
  7.4.2 Classification problem 2 97
7.5 Discussion and Conclusions 98

Appendix 102

A Some issues on least squares estimation 102
  A.0.1 The Gauss–Markov theorem 102
  A.1 Regularization theory 103

B Smoothness Priors Method for Function Estimation 104
  B.1 The smoothing problem 104
  B.2 The Shiller–Akaike Smoothing Priors Solution 105
  B.3 State space modeling 106
### List of Figures

3.1 Representation of a functional relationship .................. 16
3.2 Signal generation model ........................................ 26

5.1 Two dimensional feature problem ................................ 49
5.2 Feature extraction mapping ......................................... 50
5.3 Linear grid distribution ............................................. 51
5.4 Quadtree algorithm .................................................. 52
5.5 Feature extraction with PCA eigenfaces .......................... 55
5.6 Feature extraction with 2D–PCA .................................... 56

6.1 Experimental outline ............................................... 59
6.2 Estimation of TVAR model order ................................. 62
6.3 First TVAR parameter obtained with LS–TVAR with different window lengths $M$ .................................................. 63
6.4 First TVAR parameter obtained with SP–TVAR with different forgetting factors $\lambda$ .............................................. 63
6.5 Time–dependent variance estimates for different values of $\alpha$ ................................................................. 64
6.6 TVAR parameter estimator tuning .................................. 64
6.7 Time-dependent variance estimator tuning ....................... 65
6.8 Examples of TFR for the considered methods of estimation ... 66
6.9 Computing relevant area of TFR. Different contours appraise the changes of threshold $\delta$ .............................................. 67
6.10 Relative relevance for time and frequency axes .................. 67
6.11 Relationship between the number of divisions and segment variability ......................................................... 68
6.12 Percentage of variance explained vs. number of vectors for TFR surfaces modeled with linear decomposition methods .................. 69
6.13 First five base vectors multiplied by TFR ........................ 70
6.14 Percentage of variance explained vs. number of vectors for TVAR parameters modeled with linear decomposition methods .......... 71
6.15 Parameter relevance for model orders 4, 6 and 8 .............. 72
6.16 Number of neighbors vs. accuracy for different methods of estimation ............................................................. 73
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.17</td>
<td>Robustness assessing of tuning for tiled-TFR approach.</td>
<td>74</td>
</tr>
<tr>
<td>6.18</td>
<td>Sensibility to parameter changes for the eigenface approach.</td>
<td>75</td>
</tr>
<tr>
<td>6.19</td>
<td>Tuning of number of parameters in 2D–PCA approach.</td>
<td>75</td>
</tr>
<tr>
<td>6.20</td>
<td>Performance comparison between one dimensional and two dimensional linear decomposition approaches.</td>
<td>76</td>
</tr>
<tr>
<td>6.21</td>
<td>Model order dependence on accuracy for considered TVAR estimators.</td>
<td>76</td>
</tr>
<tr>
<td>6.22</td>
<td>Sensibility to parameter changes in final classification accuracy.</td>
<td>77</td>
</tr>
<tr>
<td>6.23</td>
<td>Sensibility of variance estimator in final classification accuracy.</td>
<td>77</td>
</tr>
<tr>
<td>6.24</td>
<td>Classification results using TVAR parameters with linear decomposition methods.</td>
<td>78</td>
</tr>
<tr>
<td>6.25</td>
<td>Number of relevant variables taken vs. classification accuracy.</td>
<td>78</td>
</tr>
<tr>
<td>6.26</td>
<td>ROC curves for feature extraction methodologies in TFR.</td>
<td>79</td>
</tr>
<tr>
<td>6.27</td>
<td>ROC curves for feature extraction methodologies in TVAR parameters.</td>
<td>80</td>
</tr>
<tr>
<td>7.1</td>
<td>Typical waveforms of database.</td>
<td>87</td>
</tr>
<tr>
<td>7.2</td>
<td>Estimation of TVAR model order for EEG signals.</td>
<td>88</td>
</tr>
<tr>
<td>7.3</td>
<td>Representative TFR sections of sets A–E.</td>
<td>89</td>
</tr>
<tr>
<td>7.4</td>
<td>Pointwise variability of TFR of EEG signals.</td>
<td>90</td>
</tr>
<tr>
<td>7.5</td>
<td>Relevance of time and frequency axes.</td>
<td>91</td>
</tr>
<tr>
<td>7.6</td>
<td>Relevance of frequency bands for different datasets.</td>
<td>91</td>
</tr>
<tr>
<td>7.7</td>
<td>Number of neighbors vs. accuracy.</td>
<td>92</td>
</tr>
<tr>
<td>7.8</td>
<td>Number of neighbors vs. accuracy using relevant area.</td>
<td>92</td>
</tr>
<tr>
<td>7.9</td>
<td>Number of components of linear decomposition vs. accuracy.</td>
<td>93</td>
</tr>
<tr>
<td>7.10</td>
<td>Number of components of two–dimensional PCA vs. accuracy.</td>
<td>93</td>
</tr>
<tr>
<td>7.11</td>
<td>Number of neighbors vs. accuracy.</td>
<td>93</td>
</tr>
<tr>
<td>7.12</td>
<td>ROC curves for estimator KS–TVAR for classification problem 1.</td>
<td>94</td>
</tr>
<tr>
<td>7.13</td>
<td>ROC curves for estimator RLS–TVAR for classification problem 1.</td>
<td>96</td>
</tr>
<tr>
<td>7.14</td>
<td>ROC curves for estimator KS–TVAR for classification problem 2.</td>
<td>97</td>
</tr>
<tr>
<td>7.15</td>
<td>ROC curves for estimator RLS–TVAR for classification problem 2.</td>
<td>98</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Time-frequency distributions and their corresponding kernel functions 34

6.1 Summary of considered datasets for classification of PCG signals . . . 72
6.2 Summary of best results for classification with parametric TFR . . . 80
6.3 Summary of best results for classification with TVAR parameters . . 80

7.1 Summary of best results for classification with parametric TFR using
KS estimator for classification problem 1 . . . . . . . . . . . . . . . . . . . 95
7.2 Summary of best results for classification with parametric TFR using
RLS estimator for classification problem 1 . . . . . . . . . . . . . . . . . . . 96
7.3 Summary of best results for classification with parametric TFR using
KS estimator for classification problem 2 . . . . . . . . . . . . . . . . . . . 97
7.4 Summary of best results for classification with parametric TFR using
RLS estimator for classification problem 2 . . . . . . . . . . . . . . . . . . . 98
7.5 Comparison of the proposed methodology with other methodologies
for classification of EEG signals . . . . . . . . . . . . . . . . . . . . . . . . . . 99
Chapter 1

Introduction

In this master’s thesis discrimination of non-stationary signals using time varying parametric modeling and time frequency analysis is explored. This work consists of two parts, the first, to obtain a representation for non-stationary signals by parametric modeling and parametric time-frequency representations, and the second, feature selection and extraction based on time–frequency representations and time-varying data.

In this study many advantages of non-stationary signal analysis using parametric methodology will be made evident. Among them it will be found that by means of these models it is possible to determine how signal’s structure changes along time and analogously, to determine how the frequency content of a signal changes.

The effectiveness of this methodology depends on three main factors, first, the choice of the model structure, which in the case of TVAR modeling would be the problem to find the order of AR model, second, estimation of the model parameters and third, selection the structure of temporal change that is imposed on the dynamics of time-variant parameters. In this aspect, a revision and evaluation of different state of the art methodologies for model structure selection, estimation of TVAR parameters and temporal structures is made. It was found that the performance of parametric methodology depends directly on these three factors; however, the main influencing factor is the structure of temporal change imposed on the estimator and how it couples with the dynamics of a time-varying signal.

The second addressed problem is how to use these time varying features (matricial features) to train classifiers. Features estimated with parametric models yield a complete representation of signal’s dynamics at the cost of large dimensionality and redundancy. Thus, a review of feature extraction methods devised for time-varying and matricial data is carried out. Also, relevance analysis is generalized for the case of matricial data.

Assessing of methodology is accomplished in classification of PCG and EEG sig-
nal signals, which exhibit different forms of non-stationarity. The proposed methodology is optimized and evaluated with different combinations of TVAR parameter estimators and feature extraction methods using nearest neighbors classifier. Results show that the proposed methodology couples appropriately to employed non-stationary signals attaining high performance on classification.

1.1 Problem statement

The classical analysis of signals is based on the assumption of stationarity of the signal studied, under which one accepts that its statistical properties are time-invariant [19]. However, this assumption doesn’t hold in most cases of real life and then it becomes necessary to use techniques that take into account the temporal variability of the signal [116]. A set of techniques known as time–frequency analysis have been proposed to tackle non–stationarity and identify the structure of change of the spectral content of a signal along time [116, 103].

Time–frequency analysis methods are often classified as non-parametric and parametric techniques [20, 98]. Non-parametric methods are based on signal representation with orthogonal or biorthogonal basis functions, with examples as the short time Fourier transform, Cohen class transforms and Wavelet transform. Its main disadvantage lies on the implicit compromise of Fourier transform with time and frequency resolution given by Heisenberg uncertainty principle [16], besides, typical appearance of missing components, typical of quadratic representations or Cohen class [114, 98].

Parametric methods are based on the assumption of a recursive model with time-dependent parameters (it time–varying auto regressive models TVAR or it time–varying auto regressive moving average models TVARMA) from which it is possible to compute the instantaneous power spectral density of the signal [116, 98]. Parametric spectral estimator has great advantages over estimates with non-parametric methods, especially because an a priori a model with a reduced number of components has been imposed on the signal, improving the estimation and eliminating undesirable effects such as dependence with resolution, cross-terms and the distortion of time–frequency plane.

However, performance of parametric models is directly linked to the assumed structure of the TVAR model, consisting of model order, parameters estimator and more importantly, the assumed form of parameter evolution over time. Depending on the form of parameters evolution different models arise, such as locally stationary parameter evolution models, adaptive parameter evolution models, stochastic parameter evolution models and deterministic parameter evolution models [98].

Locally stationary methods suppose that the signal is locally stationary and
1.1. Problem statement

regular methods such as Yule–Walker can be used to estimate parameters in short segments of length $M$ [42,92]. Adaptive methods compute estimates at each time instant $k$ based on the data available until that time in a form that is recursively updated at the next time instant $k + 1$ that the next signal sample $y[k + 1]$ is processed [55,80]. As no specific structure on the evolution of the parameters is imposed, the resulting estimates are often very noisy or too smooth. To improve the estimation of the parameters of the TVAR model it assumed that the parameters change over time according to a predefined functional form. Thus it can be assumed that the parameters evolve according to a stochastic difference equation, such as stochastic evolution models [74], or it can be assumed that the parameters vary according to a function defined by a set of basis functions, such as deterministic evolution models [38].

The previously described techniques are a form to represent non–stationary signals. However, its application for automated discrimination is not straightforward, since the amount of information which they contain is too large for a classification algorithm. Therefore it is necessary to develop methodologies to reduce the size of these features taking into account their nature as functional data which contains time and frequency relationships that must be retained. Thus, there is a growing need for new data reduction methods that can accurately parameterize the activity of time–varying features [11].

Taking into account this concept, several approaches to feature extraction from matricial data have been devised. The most rudimentary approaches consist on taking measures on matricial data, such as joint moments which are supposed to summarize information contained on TFR [115]. Therefore, by keeping all the joint moments, it’s preserved all the information in the TFR. For classification task one does not need all the joint moments of the TFR but a small subset as demonstrated in [120].

Another approach consists on inference of prototypes for each class in database and then use some distance measure from each sample in the database with the prototype. The prototype can be obtained averaging all samples from the same class, by clustering techniques (in that case it would be obtained several prototypes per class), or selecting some set of matrices that best describe remaining subjects in database [101,31].

Previous methods lack of global or local information, this is, in the case of joint moments, information of entire matrix is summarized in averages that eliminate local information; in the case of prototypes, the problem consists on correct estimation of prototypes which can generalize several conditions. So, a problem arises, how to extract features from matricial data with generalization and localization of information capability.
1.2 Objectives

1.2.1 General objective

Develop a non-stationary signal classification methodology based on parametric modeling, which can couple to signal’s time-varying dynamics and which takes into account temporal and frequential variability, improving accuracy of classifiers compared with conventional methods.

1.2.2 Specific objectives

1. Define a time-dependent parameter evolution structure that best fits to time-varying dynamics of non-stationary signals.

2. Devise a feature extraction methodology from matricial data that effectively select discriminant information obtained with time-varying parametric modeling.

3. Analyze relevance of time–frequency features and time–dependent parameters that permits assessing of time–frequency features and improves performance of feature extraction methods and classifiers.
Chapter 2

Random signals and systems

2.1 Intro

In the problem of estimating a signal $y[k]$ from the measurements $z[k] = h(y[k], v[k], k)$, the noise term $v[k]$ usually varies randomly, and thus modeling $v[k]$ requires that we use a random signal formulation. The signal $y[k]$ may also include some random variation, and thus it must be modeled in general as a random signal. The random signal formulation is generated by taking $v[k]$ and $y[k]$ as random variables for each value of the time index $k$. In the following sections we will review some concepts of random discrete time signals and linear–time varying and linear time–invariant systems driven by random signal inputs.

2.2 Summary of probability theory

In this section, a short review of notations and definitions used in this thesis is given. The fundamental definitions of probability theory will not be discussed here. These definitions can be found, e.g., in [95].

One important concept of probability theory is that of random variables. A random variable $x_j$ is a rule for assigning a value $x_j(\zeta)$ to every outcome $\zeta$ of an experiment, i.e. a function defined for all outcomes of the experiment. A random vector $\mathbf{x} = [x_1 x_2 \ldots x_N]^\top$ is a vector whose components $x_j$ are random variables. A random process $x(t)$, on the other hand, is a function of both time $t$ and the outcome $\zeta$, i.e. $x(t) = x(t; \zeta)$. For a fixed outcome $\zeta$ random process is a single function of time and for a fixed $t$ it is a random variable $x(t)$.

Let $\mathbf{x} = [x_1 x_2 \ldots x_N]^\top \in \mathbb{R}^N$ and $\mathbf{y} = [y_1 y_2 \ldots y_M]^\top \in \mathbb{R}^M$ be random vectors whose probability density functions are $p_x(\mathbf{x})$ and $p_y(\mathbf{y})$, respectively. Subscripts $x$ and $y$ of the density functions refer to the random variables and, according to a common practice, the same letters are used for the arguments of the density
functions. Therefore, without a risk of confusion the subscripts can be omitted and the density functions denoted simply as \( p(x) \) and \( p(y) \). The joint probability density of \( x \) and \( y \) is denoted as \( p(x, y) \).

The expected value or the mean \( \mu_x \) of a random vector \( x \) is defined as

\[
\mu_x = E\{x\} = \int_{\mathbb{R}^N} x p(x) \, dx
\]

where the integral is taken over each random variable \( x_j \). The mean \( E\{x\} \) is also called the first moment of \( x \). Higher moments are obtained correspondingly by integrating \( x^np(x) \). Of particular interest is the variance which is the second centered moment of \( x \) defined as

\[
\sigma_x^2 = \text{var}\{x\} = E\{(x - \mu_x)^2\}
\]

The correlation matrix of a random vector \( x \) is defined as

\[
R_x = E\{xx^\top\} = \begin{bmatrix}
E\{x_1x_1\} & \cdots & E\{x_1x_N\} \\
\vdots & \ddots & \vdots \\
E\{x_Nx_1\} & \cdots & E\{x_Nx_N\}
\end{bmatrix}
\]

and the cross-correlation matrix of random vectors \( x \) and \( y \) as

\[
R_{xy} = E\{xy^\top\}
\]

Furthermore, the covariance matrix of a random vector \( x \) is defined as

\[
C_x = E\{(x - \mu_x)(x - \mu_x)^\top\}
\]

that is the correlation matrix of the centered random vector \((x - \mu_x)\). The cross-covariance matrix of \( x \) and \( y \) is

\[
C_{xy} = E\{(x - \mu_x)(y - \mu_y)^\top\} = E\{xy^\top\} - \mu_x\mu_y^\top
\]

The conditional probability density of \( x \) given \( y \) is defined as

\[
p(x|y) = \frac{p(x, y)}{p(y)} \quad (2.1)
\]

for \( p(y) \neq 0 \), otherwise \( p(x|y) = 0 \). Likewise, the conditional density of \( y \) given \( x \) is

\[
p(y|x) = \frac{p(x, y)}{p(x)} \quad (2.2)
\]

Combining of equations (2.1) and (2.2) yields the Bayes theorem

\[
p(x|y)p(y) = p(y|x)p(x) \quad (2.3)
\]
Another useful result that can be easily derived is that for the joint conditional density of $\mathbf{x}$ and $\mathbf{y}$ given $\mathbf{z}$

$$p(\mathbf{x}, \mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{y}, \mathbf{z}) p(\mathbf{y}|\mathbf{z})$$

The conditional mean of $\mathbf{x}$ given $\mathbf{y}$ is given by

$$\mu_{\mathbf{x}|\mathbf{y}} = \mathbb{E}\{\mathbf{x}|\mathbf{y}\} = \int_{\mathbb{R}^N} \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x}$$

The components $x_j$ of a random vector are said to be jointly Gaussian if their joint probability density is of the form

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N \det C_x}} \exp \left( \frac{1}{2} (\mathbf{x} - \mu_x)^\top C_x^{-1} (\mathbf{x} - \mu_x) \right)$$

where $\det C_x$ is the determinant of $C_x$. A jointly Gaussian random vector $\mathbf{x}$ with mean $\mu_x$ and covariance $C_x$ is denoted as $\mathbf{x} \approx \mathcal{N}(\mu_x, C_x)$.

If $\mathbf{y} = L\mathbf{x}$, where $L$ is an affine transformation, then $\mathbf{y}$ is also jointly Gaussian with probability density $\mathbf{y} \approx \mathcal{N}(L\mu_x, LC_xL^\top)$.

### 2.3. Random discrete–time signals

A random discrete time signal $\mathbf{x}[k]$ is a sequence

$$\ldots, \mathbf{x}[-2], \mathbf{x}[-1], \mathbf{x}[0], \mathbf{x}[1], \mathbf{x}[2], \ldots$$

of jointly distributed random variables defined on a probability space $\mathcal{X}$. The discrete time index $k$ begins at $k = -\infty$ and ends at $k = \infty$, so that $\mathbf{x}[k]$ is a two sided random signal. An example of a one sided random signal is the sequence $\mathbf{x}[0], \mathbf{x}[1], \mathbf{x}[2], \ldots$, where $\mathbf{x}[k]$ is defined for $k \geq 0$ only.

A random discrete–time signal $\mathbf{x}[k]$ may be viewed as a sampled random continuous–time signal $\mathbf{x}(t)$. In other words,

$$\mathbf{x}[k] = \mathbf{x}(kT), \quad k = 0, \pm 1, \pm 2, \ldots$$

where $T$ is the sampling interval.

A random discrete–time signal $\mathbf{x}[k]$ can be characterized in terms of its autocorrelation function, defined by [68]

$$R_x(i, j) = \mathbb{E}\{\mathbf{x}[i], \mathbf{x}[j]\}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{x}[i]\mathbf{x}[j] p(\mathbf{x}[i], \mathbf{x}[j]) \, dx[i] dx[j]$$
2.4 Stationary Time Series

where \( p(x[i], x[j]) \) is the joint density function of the random variables \( x[i] \) and \( x[j] \). The autocorrelation is a function of two integer variables \( i \) and \( j \), with \( -\infty < i < \infty \) and \( -\infty < j < \infty \). The autocorrelation function \( R_x(i, j) \) measures the correlation between signal samples.

Suppose that the random variables comprising a random signal \( x[k] \) are independent and have means \( E\{x[k]\} = \mu_k \), where \( \mu_k \) are arbitrary nonzero real numbers. Then \( E\{x[i], x[j]\} = \mu_i \mu_j \neq 0 \) for \( i \neq j \), and thus even though \( x[i] \) and \( x[j] \) are independent (for \( i \neq j \)), and therefore unrelated \( E\{x[i], x[j]\} \) is nonzero. So there appears to be correlation between \( x[i] \) and \( x[j] \). This is a result of the nonzero mean; in fact, a nonzero mean can be interpreted as the existence of a deterministic part of the random signal \( x[k] \).

2.4 Stationary Time Series

A strictly stationary time series is one for which the probabilistic behavior of every collection of values

\[
\{x[k_1], x[k_2], \ldots x[k_N]\}
\]

is identical to that of the time shifted set

\[
\{x[k_1 + h], x[k_2 + h], \ldots x[k_N + h]\}
\]

so that

\[
p(x[k_1], x[k_2], \ldots x[k_N]) = p(x[k_1 + h], x[k_2 + h], \ldots x[k_N + h]) \tag{2.6}
\]

for all \( k = 1, 2, \ldots \) and all time shifts \( h = 0, \pm 1, \pm 2, \ldots \).

If a time series is strictly stationary, then all of the multivariate distribution functions for subsets of variables must agree with their counterparts in the shifted set for all values of the shift parameter \( h \).

The version of stationarity in (2.6) is too strong for most applications. Moreover, it is difficult to assess strict stationarity from a single data set. Rather than impose conditions on all possible distributions of a time series, we will use a milder version that imposes conditions only on the first two moments of the series. We now have the following definition.

A random discrete–time signal \( x[k] \) is said to be wide sense stationary (WSS) if the following two conditions are satisfied

\[
E\{x[k]\} = c \quad \text{for all integers } n, \tag{2.7}
\]

\[
E\{x[i]x[j]\} = E\{x[i + k]x[j + k]\} \quad \text{for all integers } i, j, k. \tag{2.8}
\]
It is very important to note that the concept of wide–sense stationarity applies to random signals $x[k]$ that are defined for $k$ ranging from $-\infty$ to $\infty$. In particular, a random signal $x[k]$ that is defined only for $k \geq 0$ only cannot be WSS.

Taking $j = i$ in condition (2.8), we see that the mean square $E\{x[k]\top x[k]\}$ must be a constant matrix for all values of $k$, and thus the variance $\text{var}\{x[k]\}$ must be a constant for all $k$. As a result, if the random variables comprising $x[k]$ are Gaussian of uniformly distributed, it turns out that $x[k]$ is WSS if and only if both the mean and the variance are constant. Unfortunately, this result is not valid in general; that is, a random signal $x[k]$ may not be WSS even though the mean and variance are constant.

If $x[k]$ is a WSS random signal, the autocorrelation function $R_x(i, j) = E\{x[i]\top x[j]\}$ is a function of the difference $i - j$. Therefore, for $x[k]$ WSS we have that $R_x[k] = E\{x[n]\top x[n - k]\}$ for any integer $n$ (2.9)

Note that $R_x[0]$ is equal to the mean square $E\{x[k]\top x[k]\}$, and thus $R_x[0]$ is always a strictly positive real number. Also note that since $R_x[-k] = E\{x[n]\top x[n - k]\}$, replacing $n$ by $n + k$ we have that

$$R_x[-k] = E\{x[n]\top x[n + k]\} = R_x[k]$$

Hence, the autocorrelation is an even function of $k$.

### 2.5 Power spectrum

Let $x[k]$ a zero mean WSS random signal with autocorrelation function $R_x[k]$. The power spectral density or power spectrum of $x[k]$, which is denoted by $S_x(e^{j\omega})$, is defined to be the discrete time Fourier transform of $R_x[k]$; that is $[68]

$$S_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_x[k] e^{j\omega k}$$

(2.10)

The function $S_x(e^{j\omega})$ represents the distribution of power with respect to frequency (where $\omega$ is the frequency variable).

Since $R_x[k]$ is an even function of $k$, it follows that $S_x(e^{j\omega})$ is real valued. Furthermore, it can be shown that $S_x(e^{j\omega}) \geq 0$ for all $\omega$, so $S_x(e^{j\omega})$ is always positive or zero. It also follows that the power spectrum is a periodic function of $\omega$ with period $2\pi$ and is symmetric about $\omega = 0$.

### 2.6 Discrete time systems with random inputs

Consider a single input single output deterministic discrete time system with deterministic input signal $w[k]$ and output signal $y[k]$. If the system is linear, the output
2.6. Discrete time systems with random inputs

Let \( y[k] \) resulting from input \( w[k] \) with no initial energy at time \(-\infty\) be given by

\[
y[k] = \sum_{i=-\infty}^{\infty} h(k; i)w[i] \tag{2.11}
\]

where \( h(k; i) \) is the output response at time \( k \) due to the application of the unit impulse at time \( i \) with no initial energy in the system prior to the application of the unit impulse. If the system is also time invariant, (2.11) reduces to the standard convolution expression

\[
y[k] = h[k] \ast w[k] = \sum_{i=-\infty}^{\infty} h(k - i)w[i] \tag{2.12}
\]

where \( h[k] \) is the impulse response.

Sample realizations of the output \( y[k] \) can be computed from sample realizations of the input \( w[k] \), nevertheless in general it is not possible to compute the distribution or density functions of random variables comprising the random signal \( y[n] \) in terms of the distribution or density functions of the random variables comprising the input random signal \( w[n] \). One exception is the case when \( w[k] \) is Gaussian distributed, the output \( y[k] \) of the system defined by (2.11) or (2.12) will also be Gaussian distributed. So in this case the output random signal is completely specified once the mean and variance have been determined. This remarkable property is true whether the system is time invariant or time varying.

### 2.6.1 Autocorrelation function of the output

Given the case for a LTI system \( h[k] \) with input \( w[k] \) WSS with mean \( \mu \) and autocorrelation function \( R_w[k] \), it can be proven that the output \( y[k] \) is also WSS, with mean

\[
\mathbb{E}\{y[k]\} = A \mu \forall n \in \mathbb{N}
\]

where

\[
A = \sum_{k=-\infty}^{\infty} h[k] \tag{2.13}
\]

and autocorrelation function

\[
R_y[k] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[-m]h(k-n)R_w[n-m] = h[k] \ast h[-k] \ast R_w[k] \tag{2.14}
\]

this last expression can be seen as an input–output relationship for LTI systems whose input is a WSS random signal. If the input is not WSS or the system is time-variant the output will not be WSS.
2.6.2 Power spectral density of the output

For a LTI system, the power spectral density $S_y(e^{j\omega})$ of the output $y[k]$ can be computed by taking the discrete time Fourier transform of both sides of (2.14). This gives

$$S_y(e^{j\omega}) = |H(e^{j\omega})|^2 S_w(e^{j\omega})$$  \hspace{1cm} (2.15)

where $H(e^{j\omega})$ is the discrete time Fourier transform of $h[k]$. If $w[k]$ is zero mean white noise with variance $\sigma^2$, then $S_w(e^{j\omega}) = \sigma^2$, and thus

$$S_y(e^{j\omega}) = |H(e^{j\omega})|^2 \sigma^2$$  \hspace{1cm} (2.16)

Hence, it is possible to determine $|H(e^{j\omega})|^2$ by measuring the power spectrum $S_y(e^{j\omega})$ of the response due to a white noise input.

2.7 Linear random process models

The most popular class of linear random process models consists of autoregressive moving average (ARMA) models, including purely autoregressive (AR) and purely moving-average (MA) models as special cases. ARMA models are frequently used to model linear dynamic structures, to depict linear relationships among lagged variables, and to serve as vehicles for linear forecasting.

A random signal $x[k]$ is called white noise, denoted as $x[k] \sim WN(0, \sigma^2)$, if

$$E\{x[k]\} = 0 \quad var\{x[k]\} = \sigma^2 \quad and \quad cov\{x[i], x[j]\} = 0 \forall i \neq j$$ \hspace{1cm} (2.17)

White noise is defined by the properties of its first two moments only. It serves as a building block in defining more complex linear time series processes and reflects information that is not directly observable. For this reason, it is often called an innovation process in the time series literature. It is easy to see that a sequence of independent and identically distributed (i.i.d.) random variables with mean 0 and finite variance $\sigma^2$ is a special white noise process. We use the notation IID $(0, \sigma^2)$ to denote such a sequence.

The probability behavior of a stochastic process is completely determined by all of its finite-dimensional distributions. When all of the finite dimensional distributions are Gaussian (normal), the process is called a Gaussian process. Since uncorrelated normal random variables are also independent, a Gaussian white noise process is, in fact, a sequence of i.i.d. normal random variables.

An autoregressive moving average (ARMA) model $x[k] \sim ARMA(p, q)$ defined as

$$x[k] = \sum_{i=1}^{p} a_i x[k-i] + \sum_{i=0}^{q} b_i \epsilon[k-i] \quad where \ \epsilon[k] \sim WN(0, \sigma^2)$$ \hspace{1cm} (2.18)


2.7. Linear random process models

$p, q > 0$ are integers, and $(p, q)$ is called the order of the model. Depending on the value of $p$ and $q$ on the ARMA$(p, q)$ model, special cases can be obtained.

An autoregressive model $x[k] = \text{AR}(p)$ of order $p \geq 1$ is defined as

$$x[k] = \sum_{i=1}^{p} a_i x[k - i] + \epsilon[k] \quad \text{where } \epsilon[k] \sim \text{WN}(0, \sigma^2) \quad (2.19)$$

Model (2.19) represents the current state $x[k]$ through its immediate $p$ past values $x[k-1], \ldots, x[k-p]$ in a linear regression form. The model is easy to implement and therefore is arguably the most popular time series model in practice. Comparing it with the usual linear regression models, we exclude the intercept in model (2.19). This can be absorbed by either allowing $\epsilon[k]$ to have a nonzero mean or deleting the mean from the observed data before the fitting. The latter is in fact common practice in time series analysis.

Model (2.19) explicitly specifies the relationship between the current value and its past values. This relationship also postulates the way to generate such an AR$(p)$ process. Given a set of initial values $x[-k_0-1], \ldots, x[-k_0-p]$, $x[k]$ can be obtained for $k \geq k_0$ iteratively from (2.19) by generating $\epsilon[k]$ from some realization of the normal distribution $\mathcal{N}(0, \sigma^2)$. Discarding the first $k_0+1$ values, we regard $\{x[k], k \geq 1\}$ as a realization of the AR process.

A moving average process $x[k] = \text{MA}(p)$ with order $q \geq 1$ is defined as

$$x[k] = \sum_{i=0}^{q} b_i \epsilon[k - i] \quad \text{where } \epsilon[k] \sim \text{WN}(0, \sigma^2) \quad (2.20)$$

An MA model expresses a time series as a moving average of a white noise process. The correlation between $x[k]$ and $x[k-h]$ is due to the fact that they may depend on the same $\epsilon[k-j]$'s. Then, $x[k]$ and $x[k-h]$ are uncorrelated when $h > q$.

Because the white noise $\epsilon[k]$ is unobservable, the implementation of an MA model is more difficult than that of an AR model. The usefulness of MA models may be viewed from two aspects. First, they provide parsimonious representations for time series exhibiting MA-like correlation structure. The second advantage of MA models lies in their theoretical tractability. It is easy to see from the representation of (2.20) that the exploration of the first two moments of $x[k]$ can be transformed to that of $\epsilon[k]$. The white noise $\epsilon[k]$ can be effectively regarded as an i.i.d. sequence when we confine ourselves to the properties of the first two moments only.
2.8 Power Spectral Density of ARMA processes

From last section, it was defined an ARMA process as

\[ x[k] = \sum_{i=1}^{p} a_i x[k-i] + \sum_{i=0}^{q} b_i \epsilon[k-i] \quad \text{where } \epsilon[k] \sim \text{WN}(0, \sigma^2) \]

Evaluating \( \mathcal{Z} \) transform in both sides, we have that

\[ X(z)A(z) = \Xi(z)B(z) \]

where \( X(z) \) is \( \mathcal{Z} \) transform of the signal \( x[k], \) \( \Xi(z) \) is the \( \mathcal{Z} \) transform of the signal \( \epsilon[k], \) and \( A(z) = 1 + \sum_{n=1}^{p} a_n z^{-n} \) and \( B(z) = 1 + \sum_{m=1}^{q} b_m z^{-m} \) are the \( \mathcal{Z} \) transforms of AR and MA polynomials. Considering \( x[k] \) as the output of a linear filter for a white noise input \( \xi[k], \) the transfer function \( H(z) \) relating input and output transforms as \( X(z) = H(z)\Xi(z) \) is

\[ X(z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 + \sum_{i=1}^{p} a_i z^{-i}} \]

or equivalently

\[ X(z) = \frac{B(z)}{A(z)} = \frac{\prod_{i=1}^{q} (z - \beta_i)}{\prod_{i=1}^{p} (z - \alpha_i)} \]

where \( \beta_m \) are the MA polynomial roots or zeros of \( H(z), \) and \( \alpha_n \) are the AR polynomial roots or poles of \( H(z). \)

\( H(z) \) belongs to a stable system if and only if all its poles lie on the unit circle, this is \( |\alpha_n| < 1 \quad \forall n = 1, \ldots, p. \) Likewise, \( H(z) \) is minimum phase if and only if all its zeros lie on the unit circle, this is \( |\beta_m| < 1 \quad \forall m = 1, \ldots, q. \)

\( \mathcal{Z} \) transform of the autocorrelation function of \( x[k] \) is related with the transform of input’s autocorrelation with

\[ R_x(z) = H(z)H(1/z)R_\xi(z) = \frac{B(z)B(1/z)}{A(z)A(1/z)}R_\epsilon(z) \]

given that the input signal \( \epsilon[k] \sim \text{WN}(0, \sigma^2), \)

\[ R_\epsilon[k] = \begin{cases} \sigma^2, & k = 0; \\ 0, & k \neq 0. \end{cases} \iff R_\epsilon(z) = \sigma^2 \]
2.9 Spectral Decomposition

Substituting \( z = e^{j2\pi f/f_s} \) yields the power spectral density of an ARMA process

\[
S_x(f) = \frac{\sigma^2}{f_s} \left| \frac{1 + \sum_{i=1}^{q} b_i e^{-j2\pi f/f_s}}{1 + \sum_{i=1}^{p} a_i e^{-j2\pi f/f_s}} \right|^2
\]

which is an estimate of the PSD if the values of the parameters \( b_i \) and \( a_i \) and variance \( \sigma^2 \) are replaced by their estimates.

Equation (2.21) is a continuous function of \( f \) and it can be evaluated for any desired frequency up to Nyquist frequency \( f_s/2 \). Nevertheless, frequency resolution is not infinite, but will be determined by the underlying model. Compared with non-parametric PSD estimation methods, the resolution of this parametric estimator is improved, due to implicit extrapolation of the autocorrelation function.

2.9 Spectral Decomposition

An important property of parametric power spectral density is that the spectrum can be split into separate components [133]. Consider a pole \( \alpha_i \) of an AR\((p)\) process placed at the frequency \( f_i \). The spectrum of this component in the neighboring region of \( f_i \) can be computed as

\[
S_i(f) \approx \frac{c_i}{(z - a_i)(z - a_i^*)}, \quad z = \exp(j2\pi f/f_s)
\]

where the constant \( c_i \) is given by

\[
c_i = \frac{\sigma^2}{f_s} \frac{1}{\prod_{n \neq i}(z - a_i)(z - a_i^*)}, \quad z = \exp(j2\pi f/f_s)
\]

This is, the part \( c_i \) of the AR spectrum is assumed constant when \( f \approx f_i \). The sum of this component’s spectrum must be approximately equal to the estimate of the AR spectrum, in other words \( S_x(f) \approx \sum_{i=1}^{p} S_i(f) \).

Power content of each component can be estimated using the method proposed in [65]. The power of the component placed in frequency \( f_i \) can be estimated with residual

\[
S_{f_i} = K \text{Re} \left\{ \text{Res} \left\{ \frac{S_x(z)}{z} \right\} \right|_{z = \exp(j2\pi f/f_s)} \right\}
\]

where the residual is evaluated in \( z = \exp(j2\pi f/f_s) \) and \( K = 1 \) for real poles and \( K = 2 \) for complex poles. Equation (2.23) can be solved computing

\[
S_{f_i} = K \text{Re} \left\{ \frac{\sigma^2(z - a_i)}{zA(z)A(z^{-1})} \right\}
\]
where \( A(z) = \prod_{i=1}^{p} (1 - \alpha_i z^{-1}) \) in \( z = \alpha_i \). This method for component's power content works well for separate components, but for close poles these estimates can lead to negative values.
Chapter 3

Estimation theory

3.1 Estimation Problem

The problem in estimation theory is to determine the parameters describing the underlying functional relationship from noisy observations. We begin from a set of observations \( \{ y_k \in \mathbb{R} \}_{k=1}^N \) related to some input \( \{ x_k = [x_{k1}, x_{k2}, \ldots x_{kp}] \in \mathbb{R}^p \}_{k=1}^N \).

The matrix \( X = [x_1 x_2 \ldots x_N]^\top \in \mathbb{R}^{N \times p} \) is used to denote the input dataset, \( y = [y_1 y_2 \ldots y_N]^\top \in \mathbb{R}^N \) the observations dataset and \( \theta = [\theta_1 \theta_2 \ldots \theta_p]^\top \in \mathbb{R}^p \) the parameters. The superscript \([\cdot]^\top\) denotes transpose. \( h_o(x_k, \theta) : \mathcal{X} \to \mathcal{Y} \) is a function which depends on the parameter vector and maps the input space \( \mathcal{X} \subset \mathbb{R}^p \) to the output space \( \mathcal{Y} \subset \mathbb{R} \). This general setting is depicted in Figure 3.1.

![Figure 3.1: Graphical representation of the functional relationship between the input space \( \mathcal{X} \) and the output space \( \mathcal{Y} \) given by the functional \( h(\cdot) \)](image)

A basic approach to solve the estimation problem is to define first a model of some specific structure describing the dependency of the observations on the model parameters and then compute the parameters in some optimal way. Such a model is here referred to as observation model.

---

1In this work we will address the estimation problem when observations are scalar, but this theory can be extended to the case of observations belonging to spaces with higher dimensions, this is \( y_k \in \mathbb{R}^q \).
3.1. Estimation Problem

The decision on model structure is usually based on some prior knowledge of the underlying system. Typically, the errors in the observations are considered additive and the observation model used is of the form

\[
y = h(X, \theta) + \varepsilon = \begin{bmatrix}
h(x_1, \theta) + \varepsilon_1 \\
h(x_1, \theta) + \varepsilon_2 \\
\vdots \\
h(x_N, \theta) + \varepsilon_N
\end{bmatrix}
\] (3.1)

where the random error \( \varepsilon \in \mathbb{R}^N \) has \( E\{\varepsilon\} = 0 \) and is independent of \( y \).

The additive error model is a useful approximation to the actual relationship. It takes into account a deterministic relationship \( h(X, \theta) \) and the influence of other unmeasured variables which contribute to \( y \), including measurement error. Most of the approximations \( h(x, \theta) \) are expressed as linear basis expansions

\[
y = h(X)\theta + \varepsilon\] (3.2)

where \( h(X) \) is a \( N \times p \) matrix (usually called observation matrix) which does not contain parameters to be estimated. This matrix is built as

\[
h(X) = \begin{bmatrix}
h_1(x_1) & h_2(x_1) & \ldots & h_p(x_1) \\
h_1(x_2) & h_2(x_2) & \ldots & h_p(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
h_1(x_N) & h_2(x_N) & \ldots & h_p(x_N)
\end{bmatrix}
\] (3.3)

and where the \( h_j \) are a suitable set of functions or transformations of an external input vector \( x_k \). This way, each a sample \( y_k \) is modeled as \( y_k = \sum_{j=1}^p h_j(x_k)\theta_j + \varepsilon_k = h(x_k)\theta + \varepsilon_k \).

Traditional examples are:

1. Linear model
   \[
y_k = x_k\theta + \varepsilon_k
   \]

2. Polynomials
   \[
y_k = \sum_{m=1}^p \sum_{n=m+1}^p \sum_{j=0}^d \theta_j x_{km}^{d-j} x_{kn}^j + \varepsilon_k
   \]

This model allows modeling of the functional relationship as a truncated Taylor series of order equal to the maximum order of the polynomial. Note, however, that the number of variables grows exponentially in the degree of the polynomial.
3.2. Bayesian estimation

In Bayesian estimation, the parameters $\theta$ are assumed to be random having a joint probability density $p(z, \theta)$ with the the observations $z = [x, y]$. The aim in Bayesian estimation is to solve the posterior density $p(\theta|z)$ of the parameters given the observations [126]. According to the Bayes’ theorem (2.3) the posterior density is

$$p(\theta|z) = \frac{p(z|\theta)p(\theta)}{p(z)} \propto p(z|\theta)p(\theta)$$

(3.4)

where $p(z|\theta)$ is the conditional density of observations $z$ given the parameters $\theta$. In Bayesian estimation $p(z|\theta)$ is called the likelihood density. The densities $p(\theta)$ and $p(z)$ are the marginal densities of the parameters $\theta$ and observations $z$, respectively. In Bayesian estimation the marginal density $p(\theta)$ is replaced with a density that describes the knowledge and/or assumptions of the parameters prior to any measurements. This replacement is not differentiated here and, thus, the marginal density $p(\theta)$ is also referred to as prior density. The marginal density of observations, given by $p(z) = \int p(z, \theta)d\theta$, on the other hand, is only a scale factor when $z$ is given.
3.2. Bayesian estimation

The posterior density (3.4) is a complete solution for the estimation problem given the observations and the prior. It assigns a value for each point \( \theta \) describing the probability of the solution for the given observations and prior. In practice, point estimates such as the mean are extracted from the posterior density and are given as solutions. The selection of the point estimates, in Bayesian estimation, is done by defining a cost function \( C(\theta, \hat{\theta}) \) that sets a unique typically real-valued cost for each combination of the true parameter values \( \theta \) and the estimated values \( \hat{\theta} \). The expected value of the cost function

\[
B(\hat{\theta}) = E\{C(\theta, \hat{\theta}(z))\} = \int_{\mathbb{R}^N} \int_{\mathbb{R}^p} C(\theta, \hat{\theta}(z)) p(z, \theta) d\theta dz
\]

is called the Bayes cost. According to the Bayes estimation criterion [126] the optimal estimator \( \hat{\theta}_B \) for the given cost function is the one that minimizes the Bayes cost, i.e.

\[
\hat{\theta}_B = \arg \min_{\hat{\theta}} B(\hat{\theta})
\]

Depending on the specific form of the cost function \( E\{C(\theta, \hat{\theta}(z))\} \) the Bayes estimate leads to specific estimators that will be explained in the following sections.

3.2.1 Mean square estimation

In the mean square (MS) estimation, the cost function is the squared norm of the estimation error \( \tilde{\theta} = \theta - \hat{\theta} \), i.e.

\[
C_{MS}(\theta, \hat{\theta}) = \|\theta - \hat{\theta}\|_2 = (\theta - \hat{\theta})^\top (\theta - \hat{\theta})
\]

The MS estimate is, thus, obtained by minimizing the Bayes cost with the given cost function. By substituting (3.6) into (3.5) the Bayes cost can be written in the form

\[
B(\hat{\theta}) = \int_{\mathbb{R}^N} \int_{\mathbb{R}^p} \|\theta - \hat{\theta}\|_2^2 p(z, \theta) d\theta dz
= \int_{\mathbb{R}^N} \left( \int_{\mathbb{R}^p} \|\theta - \hat{\theta}\|_2^2 p(\theta|z) d\theta \right) p(z) dz
\]

Given that \( p(z) \) is nonnegative, \( B(\hat{\theta}) \) is minimized by minimizing the inner integral in (3.7). This is done by taking the partial derivative of \( B(\hat{\theta}|z) \) with respect to \( \hat{\theta} \) and setting it to zero. This yields

\[
\frac{\partial B(\hat{\theta}|z)}{\partial \hat{\theta}} = \int_{\mathbb{R}^p} \frac{\partial}{\partial \hat{\theta}} \left( \theta^\top \theta - 2\theta^\top \hat{\theta}_{MS} + \hat{\theta}_{MS}^\top \hat{\theta}_{MS} \right) p(\theta|z) d\theta
= \int_{\mathbb{R}^p} \left( 2\hat{\theta}_{MS} - 2\theta \right) p(\theta|z) d\theta = 0
\]

May 21, 2009
3.2. Bayesian estimation

Since $\hat{\theta}_{MS}$ does not depend on $\theta$, it can be taken outside the integral and the previous equation can be rearranged to yield

$$\hat{\theta}_{MS} \int_{\mathbb{R}^p} p(\theta | z) d\theta = \int_{\mathbb{R}^p} \theta p(\theta | z) d\theta$$

Since the integral over the conditional density is naturally equal to unity, it is concluded that

$$\hat{\theta}_{MS} = \int_{\mathbb{R}^p} \theta p(\theta | z) d\theta = \mu_{\theta|z}$$

i.e. the MS estimator is equal to the conditional mean of parameters $\theta$ given the observations $z$. Based on this, the estimator is sometimes also called as the conditional mean estimator. Furthermore, it could be easily shown that the expected value of the estimation error $\tilde{\theta} = \theta - \hat{\theta}_{MS}$ is zero [126] and, therefore, the MS estimate is unbiased. Because $\tilde{\theta}$ is zero mean, the estimation error variance is of the form

$$\text{var}\{\tilde{\theta}\} = E\{\tilde{\theta}^\top\tilde{\theta}\} = E\{(\theta - \hat{\theta}_{MS})^\top(\theta - \hat{\theta}_{MS})\}$$

and, thus, the conditional mean minimizes the variance of the estimation error. Therefore, $\hat{\theta}_{MS}$ is also called minimum error variance estimator or just minimum variance estimator.

3.2.2 Maximum a posteriori estimation

Another possible cost function is the uniform cost (UC) function given by

$$C_{UC}(\theta, \tilde{\theta}) = \begin{cases} 0, & \text{if } |\tilde{\theta}_k| < \epsilon, \forall k, \\ 1, & \text{otherwise.} \end{cases}$$

where $\epsilon$ is a small constant. This cost function gives zero penalty if all components of the estimation error $\tilde{\theta}$ are small and unit penalty if any of the components is larger than $\epsilon$. By substituting this cost function into (3.5) results

$$B(\tilde{\theta}) = \int_{\mathbb{R}^p} p(\theta | z) d\theta - p(\tilde{\theta} | z) \epsilon = 1 - p(\tilde{\theta} | z) \epsilon$$

The estimate which now minimizes the Bayes cost is called the maximum a posteriori (MAP) estimate

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \left\{ p(\tilde{\theta} | x) \right\}$$

(3.8)

This solution equals the mode (maximum) of the posterior probability. It can also be written entirely in terms of the prior probability densities and the conditional probabilities:

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \left\{ \frac{p(z | \theta) p(\theta)}{p(z)} \right\} = \arg \max_{\theta} \left\{ p(z | \theta) p(\theta) \right\}$$

(3.9)
In other words, $\hat{\theta}_{MAP}$ is the mode of the conditional density $p(\theta | z)$. Due to these results, the estimator $\hat{\theta}_{MAP}$ is called the conditional mode estimator.

A useful equivalence can be easily observed. That is, if the posterior density $p(\theta | z)$ is symmetric and unimodal then the mode and the mean of the density function are the same. This means that the MS and MAP estimates are the same, i.e. $\hat{\theta}_{MS} = \hat{\theta}_{MAP}$.

### 3.2.3 Maximum likelihood estimation

In many practical situations the prior knowledge needed in MAP estimation is not available. In these cases, an estimator which does not depend on prior knowledge is desirable. One attempt in that direction is the method referred to as maximum likelihood estimation (ML estimation). The method is based on the observation that in MAP estimation (3.9), the peak of the first factor $p(z | \theta)$ is often in an area of $\theta$ in which the second factor $p(\theta)$ is almost constant. This holds true especially if little prior knowledge is available. In these cases, the prior density $p(\theta)$ does not affect the position of the maximum very much. Discarding the factor, and maximizing the function $p(z | \theta)$ solely, gives the ML estimate:

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \left\{ p \left( z | \hat{\theta} \right) \right\} \quad (3.10)$$

Regarded as a function of $\theta$ the conditional probability density is called the likelihood function. Hence the name maximum likelihood estimation.

Another motivation for the ML estimator is when we change our viewpoint with respect to the nature of the parameter vector $\theta$. In the Bayesian approach $\theta$ is a random vector, statistically defined by means of probability densities. In contrast, we may also regard $\theta$ as a non random vector whose value is simply unknown. This is the so-called Fisher approach. In this view, there are no probability densities associated with $\theta$. The only density in which $\theta$ appears is $p(z | \theta)$, but here $\theta$ must be regarded as a parameter of the density of $z$. From all estimators discussed so far, the only estimator that can handle this deterministic point of view on $\theta$ is the ML estimator.

### 3.3 Least squares estimation

In this section, will be presented the least squares (LS) estimate of the parameters $\theta$ and error term $\varepsilon$ in the linear observation model:

$$y = X\theta + \varepsilon$$
3.3. Least squares estimation

In the LS estimation, neither the parameters $\theta$ nor the observation error $\varepsilon$ are interpreted as random variables, but the LS estimation can be considered as a deterministic fit. The LS solution is defined as the vector $\theta$ that minimizes the variance of error, i.e. the minimizer of the function

$$l(\theta) = E\{(y - x^T \theta)^2\}$$

(3.11)

this function is minimized when its derivative is zero, so

$$\frac{\partial}{\partial \theta} E\{(y - x^T \theta)^2\} = -2E\{xy\} + 2E\{xx^T\} \theta = 0$$

(3.12)

where $E\{xy\} = C_{xy}$ can be recognized as the covariance matrix of $x$ and $y$, and $E\{xx^T\} = C_{xx}$ the autocovariance matrix of $x$. Solving the equation for $\theta$ yields

$$\hat{\theta} = C_{xx}^{-1}C_{xy}$$

(3.13)

which gives a closed solution for the parameter estimation problem in the linear observation model, but requires knowledge of covariance matrices $C_{xx}$ and $C_{xy}$. When these quantities are unknown, they must be estimated as well. The empirical risk minimization principle gives a light on how this estimation process can be accomplished.

### 3.3.1 Empirical risk minimization principle

Up to this point, we have considered the estimation problem as that of minimizing a risk or cost function supposing we have a priori knowledge of the data distribution $p(z, \theta)$. In most of the cases this information is unavailable, but a finite set of measurements (empirical data) can be used to compute the estimate $\hat{\theta}$.

To minimize the risk functional based on empirical data, instead of minimizing the risk functional (3.5), the empirical risk functional can be used

$$B_{emp}(\theta) = \frac{1}{N} \sum_{i=1}^{N} C(\theta, \hat{\theta})$$

(3.14)

Let the minimum of the risk functional be attained at $C(\theta, \hat{\theta}_{opt})$ and let the minimum of the empirical risk functional be attained at $C(\theta, \hat{\theta}_{emp})$. The function $C(\theta, \hat{\theta}_{emp})$ is considered as an approximation of the function $C(\theta, \hat{\theta}_{opt})$. This principle of solving the risk minimization problem is called the empirical risk minimization (induction) principle ERM [128].

The ERM principle is quite general. The classical methods for the solution of a specific learning problem, such as the least-squares method are realizations of
the ERM principle for specific loss functions. Indeed, in the case of mean square estimator, the empirical risk takes the form

$$B_{emp}(\hat{\theta}) = \frac{1}{N} \sum_{i=1}^{N} (y - X\theta)^2$$

(3.15)

### 3.3.2 Least squares estimation from experimental data

The (empirical) least squares estimate can be carried out in a similar way, by minimizing equation (3.15), which leads to

$$\hat{\theta}_{LS} = (X^\top X)^{-1} X^\top y$$

(3.16)

Notice that $X^\top X$ is an estimate of the covariance matrix $C_{xx}$, whereas $X^\top y$ is an estimate of the covariance matrix $C_{xy}$, in both cases making use of experimental data. This is a unique solution of the function (3.11) if the matrix $X^\top X$ is positive definite. The LS estimate for the observations is, furthermore, obtained as

$$\hat{y}_{LS} = X\hat{\theta}_{LS}$$

Note, that the residual $\varepsilon = y - \hat{y}_{LS}$ is orthogonal to $\mathbb{R}(H)$ and $\hat{\theta}_{LS}$ is simply the orthogonal projection of $y$ onto $\mathbb{R}(X)$.

The Gauss-Markov theorem (Appendix A) states that the least squares estimate of the parameter vector $\theta$ have the smallest variance among all linear unbiased estimates. Nevertheless this property is not always enough to prefer LS estimate, and other estimates with some bias but smaller variance might be preferred. Besides, when the matrix $(X^\top X)^{-1}$ is singular or near singular, LS estimates don’t exist. In these cases, inclusion of a regularization term in (3.11) can improve estimates.

### 3.4 Recursive estimation

Another form to solve the estimation problem is to compute estimates as new data becomes available. This is useful in the case of sequential data such as time series. In this setting, estimation of covariance matrices $C_{xx}$ and $C_{xy}$ with batch least squares improves as more data becomes available, but computation effort grows as well. Recursive estimation bridges this problem computing estimates as new data becomes available, making use of past information stored in the previously computed estimates. In the resting part of the chapter the most common recursive algorithms for parameter estimation will be reviewed.
3.4. Recursive estimation

3.4.1 Least mean square algorithm

As was derived in previous sections, the optimal least squares solution \( \hat{\theta} = C_{xx}^{-1}C_{xy} \) is obtained by finding the minimum of the least squares risk function (3.11)

\[
l(\theta) = E\{(y - x^T \theta)^2\} \tag{3.17}
\]

The optimal estimate can also be obtained using a steepest descent algorithm as follows [33]

\[
\theta[k + 1] = \theta[k] - \mu g_\theta[k] \tag{3.18}
\]

and iterating until convergence is achieved. \( \mu g_\theta[k] \) is the derivative of (3.11) with respect to the parameter vector \( \theta \):

\[
g_\theta[k] = -2C_{xy} + C_{xx}\theta[k]
\]

Given that the specific values of \( C_{xy} \) and \( C_{xx} \) are unknown, approximate values should be used. In the case of LMS algorithm, these values are approximated as follows

\[
C_{xx}[k] = x[k]x^T[k], \quad C_{xy}[k] = x^T[k]y[k]
\]

so the estimate of \( g_\theta[k] \) is

\[
\hat{g}_\theta[k] = -2y[k]x[k] + 2x[k]x[k]^T \theta[k] = -2x[k](y[k] - x[k]^T \theta[k])
\]

\[
= -2e[k]x[k]
\]

where \( e[k] = y[k] - x[k]^T \theta[k] \) is the instantaneous error. Using this estimate, the update equation in (3.19) becomes

\[
\theta[k + 1] = \theta[k] + 2\mu e[k]x[k] \tag{3.19}
\]

The parameter vector can be initialized in zeros or in some value near the optimal value to reduce the number of iterations to reach the neighborhood of \( \theta_{LS} \).

3.4.2 Recursive Least Squares Algorithm

Least squares algorithms aim at the minimization of the sum of the squares of the difference between the desired signal and the model output. When new samples are received at every iteration, the solution for the least squares problem can be computed in recursive form resulting in the recursive least squares (RLS) algorithm.

To do so, the objective function is rewritten as

\[
l(\theta) = \sum_{i=0}^{k} \lambda^{k-i} e^2[i] = \sum_{i=0}^{k} \lambda^{k-i} (y[i] - x^T[i] \theta[k])^2
\]

\[
= \sum_{i=0}^{k} \lambda^{k-i} (y[i] - \theta[i])^2
\]

May 21, 2009
differentiating \( l(\theta) \) with respect to \( \theta \), it follows that

\[
\frac{\partial l(\theta)}{\partial \theta} = -2 \sum_{i=0}^{k} \lambda^{k-i} x[i] (y[i] - x^\top[i] \theta[k])
\]

it is possible to obtain the optimal value by equating to zero and solving for \( \theta \),

\[
-2 \sum_{i=0}^{k} \lambda^{k-i} x[i] (y[i] - x^\top[i] \theta[k])^2 = 0
\]

so the optimal value is obtained at

\[
\theta[k] = \left( \sum_{i=0}^{k} \lambda^{k-i} x[i] x^\top[i] \right)^{-1} \sum_{i=0}^{k} \lambda^{k-i} x[i] y[i]
\]

where \( \hat{C}_{xx}[k] \) and \( \hat{C}_{xy}[k] \) are approximate estimates of the covariance matrices using information up to time \( k \). Notice that these matrices can be rewritten as

\[
\hat{C}_{xx}[k] = x[k] x^\top[k] + \lambda \hat{C}_{xx}[k-1]
\]
\[
\hat{C}_{xy}[k] = x[k] y[k] + \lambda \hat{C}_{xy}[k-1]
\]

using the matrix inversion lemma, the inverse \( \hat{C}_{xx}^{-1}[k] \) can be written as

\[
\hat{C}_{xx}^{-1}[k] = S_{xx}[k] = \frac{1}{\lambda} \left( S_{xx}[k-1] - \frac{S_{xx}[k-1] x[k] x^\top[k] S_{xx}[k-1]}{\lambda + x^\top[k] S_{xx}[k-1] x[k]} \right) \tag{3.21}
\]

and the recursive least squares solution is given by

\[
\theta[k] = S_{xx}[k] \left( x[k] y[k] + \lambda \hat{C}_{xy}[k-1] \right) \tag{3.22}
\]

Equations (4.16) and (3.21) define the recursive least squares algorithm. This algorithm is robust and easy to implement. Also should be noted that the perturbation is white noise. The RLS algorithm can be initialized with the values \( \theta[0] = E\{\theta\} \) and \( S_{xx}[k] = \delta I \), where \( \delta \) is some large constant.

### 3.4.3 Recursive Bayesian Estimation

In this section we review some ideas around the problem of state estimation in discrete linear systems which can be described as dynamic state space models (see Fig. 3.2). The system hidden state \( x[k] \) evolves over time \( k \) according to a first order Markov process with a conditional probability density \( p(x[k]|x[k-1]) \) and initial distribution \( p(x_0) \). Observations from the process \( y[k] \) are generated according with
3.4. Recursive estimation

the probability density \( p(y[k]|x[k]) \). The dynamic model can also be described in an equivalent form with the set of equations

\[
\begin{align*}
    x[k] &= f(x[k-1], v[k]) \tag{3.23} \\
    y[k] &= h(x[k], n[k]) \tag{3.24}
\end{align*}
\]

where \( v[k] \) is the innovations sequence that gives randomness to the state transition function \( f(\cdot) \), and \( n(k) \) is observation noise given by the measurement system through the observation function \( h(\cdot) \). The state transition probability density function \( p(x[k]|x[k-1]) \) is totally specified by \( f(\cdot) \) and the probability density function \( p(v[k]) \), whereas \( h(\cdot) \) and observation noise distribution \( p(n[k]) \) fully describe the observation probability \( p(y[k]|x[k]) \).

![Figure 3.2: Signal generation model](image)

Sequential probabilistic inference is the problem of estimating the hidden states of a system in an optimal and consistent fashion as a set of incomplete or noisy measurements become available [127]. This is, given the measurements \( \{y[k]\}_{k=1}^{N} \), the estimation problem is that of finding the estimated values \( \{\hat{x}[k]\}_{k=1}^{N} \) of the state, optimal in some sense. All the statistical information contained in the data \( \{y[k]\}_{k=1}^{N} \) about the signal is contained on the conditional probability density of the state given the noisy measurements, \( p(x[k]|y[1]...y[k]) = p(x[k]|y_1^k) \), where \( y_1^k = [y[1]...y[k]] \).

From a Bayesian viewpoint, the filtering a posteriori density \( p(x[k]|y_1^k) \) of the state given all the observations up to time \( k \) constitutes the complete solution for the probabilistic inference problem, which let compute any optimal estimate, such as the conditional mean

\[
\hat{x}[k] = E\{x[k]|y_1^k\} = \int x[k]p(x[k]|y_1^k) \, dx[k]
\]

The optimal method to recursively update the a posteriori density as new information becomes available is by means of the recursive Bayesian estimation algorithm. This method first projects the previous a posteriori density \( p(x[k]|y_1^{k-1}) \) to
the next time instant using the probabilistic model of the process,

\[ p(x[k]|y_1^{k-1}) = \int p(x[k]|x[k-1]) p(x[k-1]|y_1^{k-1}) \, dx[k-1] \]  

(3.25)

and then incorporates the last noisy measurement using the observation probability to generate the updated a posteriori density

\[ p(x[k]|y_k) = C p(y[k]|x[k]) p(x[k]|y_1^{k-1}) \]  

(3.26)

where the normalization factor \( C \) is given by

\[ C = \left( \int p(y[k]|x[k]) p(x[k]|y_1^{k-1}) \, dx[k] \right)^{-1} \]

Though this is the optimal recursive solution, a closed form for these integrals can only be found in the case of Gaussian linear systems, in which case, the solution is known as the Kalman filter [67]. Nevertheless, for the most non-linear non-Gaussian systems in real life, approximate solutions should be used [121].

### 3.4.4 Kalman filter

In the linear case, equations (3.23) and (3.24) can be written as:

\[ x[k] = F[k]x[k-1] + v[k] \]  

(3.27)

\[ y[k] = H[k]x[k] + n[k] \]  

(3.28)

it should be noted that even though the system is linear, it can be time–varying, so the matrices \( F[k] \) and \( H[k] \) can be time dependent.

We will review the demonstration of Kalman filter equations from the MAP estimation point of view as given in [91], which gives better insight into the relationship of the evolution of probability densities and the Kalman filter. The demonstration from the MSE estimation point of view can be found for example in [50,56].

The sequential MAP estimation selects the current estimate \( \hat{x}[k] \) which is more likely given the model and the observations \( \{y(k)\}^k \). This objective can be posed formally as

\[ \hat{x}[k] = \arg \max_{x[k]} p(x[k]|y_1^k) \]  

(3.29)

The joint probability function to be maximized can be rewritten as

\[ p(x[k]|y_1^k) = \frac{p(x[k]|y_1^k)}{p(y_1^k)} = \frac{p(x[k]|y[k]|y_1^{k-1}) \, p(y_1^{k-1})}{p(y_1^k)} \]  

(3.30)
Given that \( p\left(y_{i}^{k-1}\right)\) and \( p\left(y_i^{k}\right)\) are independent of \( x[k] \), the MAP estimate can be obtained only maximizing the term \( p\left(x[k]y[k]|y_{i}^{k-1}\right) \). This term can be expanded as

\[
p\left(x[k]y[k]|y_{i}^{k-1}\right) = p\left(y[k]|y_{i}^{k-1}x[k]\right) p\left(x[k]|y_{i}^{k-1}\right)
\]

(3.31)

In this case \( p\left(y[k]|y_{i}^{k-1}x[k]\right) = p\left(y[k]|x[k]\right) \). If the process noise \( v[k] \) and the measurement noise \( n[k] \) are zero mean Gaussian, then the densities in Eq.(3.31) are

\[
p\left(y[k]|x[k]\right) = \frac{1}{\sqrt{(2\pi)^N|R_n|}} \exp\left[-\frac{1}{2}(y[k] - Cx[k]) (R_n)^{-1} (y[k] - Cx[k])^\top\right]
\]

\[
p\left(x[k]|y_{i}^{k-1}\right) = \frac{1}{\sqrt{(2\pi)^M|P_x^{-}[k]|}} \exp\left[-\frac{1}{2}(x[k] - \hat{x}^{-}[k]) (P_x^{-}[k])^{-1} (x[k] - \hat{x}^{-}[k])^\top\right]
\]

(3.32)

where \( N \) is the dimension of \( y[k] \), \( M \) is the dimension of \( x[k] \), and

\[
\hat{x}^{-}[k] = E\{x[k]|y_{i}^{k-1}\}
\]

\[
P_x^{-}[k] = E\{(x[k] - \hat{x}^{-}[k]) (x[k] - \hat{x}^{-}[k])^\top|y_{i}^{k-1}\}
\]

are the a priori mean and covariance of \( x[k] \) given the data \( y_{i}^{k-1} \).

Taking the negative logarithm, results the MAP cost function

\[
J(x[k]) = D + \frac{1}{2}(y[k] - Cx[k]) (R_n)^{-1} (y[k] - Cx[k])^\top \ldots
\]

\[
\ldots + \frac{1}{2}(x[k] - \hat{x}^{-}[k]) (P_x^{-}[k])^{-1} (x[k] - \hat{x}^{-}[k])^\top
\]

(3.33)

where \( D \) is a constant which carries the Gaussian probability densities normalization terms. So, the MAP estimate \( \hat{x}[k] \) can be obtained minimizing the expression in Eq.(3.33).

Taking the derivative of (3.33) with respect to \( x[k] \) and equating to zero yields

\[
\frac{\partial J(x[k])}{\partial x[k]} = -C^\top(R_n)^{-1} (y[k] - Cx[k]) + (P_x^{-}[k])^{-1} (x[k] - \hat{x}^{-}[k])
\]

\[
= (P_x^{-}[k])^{-1} (x[k] - \hat{x}^{-}[k]) \ldots
\]

\[
\ldots - C^\top(R_n)^{-1} [y[k] - C(x[k] - \hat{x}^{-}[k]) - C\hat{x}^{-}[k]] = 0
\]

(3.34)

Rejoining terms containing \( (x[k] - \hat{x}^{-}[k]) \) yields

\[
[(P_x^{-}[k])^{-1} + C^\top(R_n)^{-1}C] (x[k] - \hat{x}^{-}[k]) = C^\top(R_n)^{-1} (y[k] - C\hat{x}^{-}[k])
\]

(3.35)

and solving for \( x[k] \) results

\[
x[k] = \hat{x}^{-}[k] + [(P_x^{-}[k])^{-1} + C^\top(R_n)^{-1}C]^{-1} C^\top(R_n)^{-1} (y[k] - C\hat{x}^{-}[k])
\]

(3.36)
3.4. Recursive estimation

which may be rewritten as

\[
\mathbf{x}[k] = \hat{\mathbf{x}}^{-}[k] + \mathbf{K}[k] (\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k]) \tag{3.37}
\]

\[
\mathbf{K}[k] \triangleq \left[ (\mathbf{P}_x^{-}[k])^{-1} + C^\top (R_n)^{-1} C \right]^{-1} C^\top (R_n)^{-1} \tag{3.38}
\]

where the quantity \( \mathbf{K}[k] \) is known as Kalman gain. During computation of Kalman gain inversion of a \( M \times M \) matrix should be carried out. In order to diminish computational cost on this matrix inversion we can make use of matrix inversion lemma which gives us a form to change the inversion of a \( M \times M \) matrix by an inversion of a matrix with the dimensions of the observation vector \( \mathbf{y}[k] \), which in most of the cases has lower dimension than the state vector. In this form, we obtain

\[
\mathbf{K}[k] = \mathbf{P}_x^{-}[k] C^\top (C \mathbf{P}_x^{-}[k] C^\top + R_n)^{-1} \tag{3.39}
\]

A posteriori error covariance update

The a posteriori error covariance matrix \( \mathbf{P}_x[k] \) is computed using its definition:

\[
\mathbf{P}_x[k] = \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}[k])(\mathbf{x}[k] - \hat{\mathbf{x}}[k])^\top\} \tag{3.40}
\]

substituting into (3.37) yields:

\[
\mathbf{P}_x[k] = \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k] - \mathbf{K}[k](\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k]))
\]

\[
(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k] - \mathbf{K}[k](\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k]))^\top\} \tag{3.41}
\]

\[
\begin{align*}
\mathbf{P}_x[k] &= \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])^\top\} \\
&\quad - \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])(\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k])^\top\} \mathbf{K}[k] \\
&\quad - \mathbf{K}[k] \mathbf{E}\{(\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k])(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])^\top\} \\
&\quad + \mathbf{K}[k] \mathbf{E}\{(\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k])(\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k])^\top\} \mathbf{K}[k] \tag{3.42}
\end{align*}
\]

Whereas the first term in Eq.(3.42) can be evaluated as \( \mathbf{P}_x^{-}[k] \), evaluation of the other terms requires rewrite the term \( \mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k] \) as

\[
\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k] = C\mathbf{x}[k] + \mathbf{n}[k] - C\hat{\mathbf{x}}^{-}[k]
\]

\[
= C(\mathbf{x}[k] + \hat{\mathbf{x}}^{-}[k]) - \mathbf{n}[k] \tag{3.43}
\]

so that the second term in Eq.(3.42) becomes

\[
\begin{align*}
\mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])(\mathbf{y}[k] - C\hat{\mathbf{x}}^{-}[k])^\top\} \\
&= \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])^\top C^\top\} + \mathbf{E}\{(\mathbf{x}[k] - \hat{\mathbf{x}}^{-}[k])\mathbf{n}^\top[k]\} \\
&= \mathbf{P}_x^{-}[k] C^\top \tag{3.44}
\end{align*}
\]
here the cross term vanishes given that the measurement error \( n[k] \) is assumed to be Gaussian and uncorrelated with \( x[k] - \hat{x}^- [k] \). The third term is the transpose of the second and the fourth contains;

\[
E\{(y[k] - C\hat{x}^- [k])(y[k] - C\hat{x}^- [k])^T\} = CE\{(x[k] - \hat{x}^- [k])(x[k] - \hat{x}^- [k])^T\} C^T + CE\{(x[k] - \hat{x}^- [k])n^T [k]\} + \cdots \]

\[
= CP_x[k]C^T + R_n
\]

(3.45)

where once again the cross terms vanish. Substituting Eq.(3.44) and Eq.(3.45) in Eq.(3.42) we get

\[
P_x[k] = P_x^- [k] - P_x^- [k]C^T K^T [k] - K[k]CP_x^- [k] + K[k] (CP_x^- [k]C^T + R_n) K[k]
\]

using \( P_x^- [k]C^T (CP_x^- [k]C^T + R_n)^{-1} \), we obtain

\[
P_x[k] = P_x^- [k] - K[k]CP_x^- [k] + P_x^- [k]C^T K^T [k] = (I - K[k]C)P_x^- [k]
\]

(3.46)

In this way we found an equation for a posteriori error covariance \( P_x[k] \) as a linear function of a priori error covariance \( P_x^- [k] \).

**A priori state and error covariance update equations**

Finally, we must find the relationships to update the values of \( \hat{x}^- [k+1] \) and \( P_x^- [k+1] \).

Using the state space representation, the obtention of this values is straightforward. The a priori state estimate is:

\[
\hat{x}^- [k + 1] = E\{x[k + 1]|y^k_i\}
\]

\[
= E\{Ax[k] + v[k]|y^k_i\}
\]

\[
= AE\{x[k]|y^k_i\} + E\{v[k]|y^k_i\}
\]

\[
= A\hat{x}^- [k]
\]

(3.47)

where the conditional expectation of \( v[k] \) is zero under supposition that the noise is white.

The a priori covariance is obtained as:

\[
P_x^- [k + 1] = E\{(x[k + 1] - \hat{x}^- [k + 1])(x[k + 1] - \hat{x}^- [k + 1])^T|y^k_i\}
\]

\[
= E\{(Ax[k] + v[k] - A\hat{x}^- [k])(Ax[k] + v[k] - A\hat{x}^- [k])^T|y^k_i\}
\]

\[
= AE\{(x[k] - \hat{x}^- [k])(x[k] - \hat{x}^- [k])^T|y^k_i\} A^T + E\{v[k]v[k]^T|y^k_i\}
\]

\[
= AP_x^- [k]A^T + R_v
\]

(3.48)
These equations to update the a priori mean and covariance from a posteriori mean and covariance are usually known as *temporal update equations* of the Kalman filter. The equations to update the a posteriori values are known as *measurement correction equations*.
Chapter 4

Non–stationary signal models

4.1 Introduction

Non-stationary random signals are characterized by time-dependent statistical moments. Confining attention to the first two moments — which completely define the probability distribution in the Gaussian case — the mean (1st moment) and autocovariance (2nd moment) are of the following respective forms [98]:

\[ \mu_x[k] = E\{x[k]\} = \int_{-\infty}^{\infty} x[k] p(x[k]) \, dx[k] \] \hspace{1cm} (4.1)

\[ \text{cov}\{k_1, k_2\} = E\{x[k_1]x[k_2]\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x[k_1]x[k_2] p(x[k_1], x[k_2]) \, dx[k_1] dx[k_2] \] \hspace{1cm} (4.2)

Unlike in the stationary case, the mean is, in general, a function of time, and the covariance a function of two considered time instants. It is oftentimes convenient to think of the covariance as being of a local nature, relating values of the signal around a time instant \( k \). In that case \( \text{cov}\{\cdot, \cdot\} \) is typically expressed as

\[ \text{cov}\{k - l, k + l\} = E\{x[k - l]x[k + l]\} \\
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x[k - l]x[k + l] p(x[k - l], x[k + l]) \, dx[k - l] dx[k + l] \]

We will focus on Gaussian zero-mean random signals with non-stationary and continually evolving covariance function. The zero-mean assumption is adopted because in most applications the mean is either zero or constant (independent of time). Whenever this assumption doesn’t hold, mean may be (in an initial stage) estimated and subsequently subtracted from the signal. The case of a time-dependent mean
(also referred to as a deterministic trend function) may be also treated in an initial stage via proper techniques, such as curve fitting or high pass signal filtering. The continual evolution assumption for the covariance function is adopted because this is the most common setting in non-stationary signals, where a continual evolution of the dynamics is encountered.

Non-stationary random signal analysis has been an important tool in biosignal analysis; see for example [117, 118, 107, 64, 63, 36, 27], etc. The span of methods may be classified as parametric or non-parametric. In the following sections we will review parametric and non parametric methods, but with more attention on parametric methods, which are of particular interest on this work.

4.2 Non-parametric approach

Non-parametric methods of time-varying spectral analysis are based upon non-parameterized representations of energy as a simultaneous function of time and frequency, known as time-frequency distributions (TFD). Thus, based on the expansion and inner product concepts, a direct way of describing a signal in $t-f$ domains consists on their comparison to elementary functions that are compacted in $t-f$ plane. In this scope, and grounded on classical Fourier Transform, $\mathcal{F}_f\{x(t)\}$, the STFT introduces a time localization concept by using a tapering window function $\varphi(t)$ going along the signal $x(t)$. Since the location of the sliding window adds a time dimension, this linear TFR is accomplished as follows:

$$S_{\varphi}(f) = \langle x, \varphi \rangle = \int_t x(t)\varphi_a,b(f,t)dt$$

(4.3)

where $T \in \mathbb{R}, \varphi_{a,b}(f,t) \in L^1(\mathbb{R}), x \in L^1(\mathbb{R}), (a,b) \in \mathbb{R}^2$.

The windowing function must be symmetrical and normalized; hence, the $t-f$ atom is defined as:

$$\varphi_{a,b}(t) = e^{-2\pi jft}\varphi(t - b)$$

(4.4)

which gives a relationship between the signal, $x(t)$, and a sort of functions with the energy compacted in narrow strips of the $t-f$ plane. The $t-f$ density of $x(t)$ can be represented by means of the spectrogram:

$$|S_{\varphi}(t,f)|^2 = \left| \int_T x(\tau)\varphi(\tau - t)e^{-2\pi jf\tau}d\tau \right|^2, \quad t, \tau \in T$$

(4.5)

In the STFT the window length remains constant. Therefore, the extraction of information with fast changes in time (i.e. high frequency components), must be accomplished with short and well-timed localized intervals, but not over the whole

May 21, 2009
4.2. Non-parametric approach

interval of definition of the PCG signal. And vice versa, low frequency components involve large time intervals of analysis. As a result, being a nonstationary signal of relatively small time-bandwidth product, TFR based on STFT is not enough suitable of revealing the $t-f$ dynamics. Certainly, this issue is partially solved using different windowing functions [23].

In practice, quadratic energy distributions, which distribute the energy of a signal over $t-f$ planes without windowing, are broadly used because of their flexibility, since time and frequency resolution can be adapted independently to suit the particular signal and cross terms [83]. One of the most commonly studied TFD with high resolution is the generalized bilinear class, or distributions of Cohen’s class, $X_C(t,f)$ defined for time $t$ and frequency $f$ as follows:

$$S_C(t,f) = \int_T h_C(t,\tau)x(t-\tau/2)x^*(t+\tau/2)e^{-2\pi jf\tau}d\tau \quad (4.6)$$

where $X_C(t,f) \in L^2(\mathbb{R})$ and the 2-D function $h_C(t,\tau) \in L^2(\mathbb{R})$ is a time-lag kernel which defines the particular TFD. The variety of TFD and their properties are determined by the choice of the kernel functions. For instance, by setting the time-lag kernel equal to 1; that distribution will be recognized as the Wigner–Ville Distribution (WVD) which provides high resolution in both time and frequency for a mono component signals. WVD satisfies some desirable properties of TFD, such as positiveness, time and frequency marginal and Kernel continuity on marginal axis origin [25], but for multi-component signals generates cross-term artifacts, rendering poor performance of this TFR. Several approaches were proposed to deal with this phenomenon, using different kernel functions which try to smooth cross–terms. Some TFD and their respective kernels are shown in Table 4.1 [82].

<table>
<thead>
<tr>
<th>TFD</th>
<th>Kernel $h_C(t,\tau)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wigner–Ville (WVD)</td>
<td>1</td>
</tr>
<tr>
<td>Choi–Williams (CWD)</td>
<td>$(4\pi \tau^2)/(\sigma^{-1/2}) \exp(-\sigma t^2/4\tau^2)$</td>
</tr>
<tr>
<td>Exponential T-distribution (ETD)</td>
<td>$(\sigma/\pi)^{-1/2} \exp(-\sigma t^2)$</td>
</tr>
<tr>
<td>Hyperbolic T-distribution (HTD)</td>
<td>$\Gamma(2\sigma)/(2^{2\sigma-1}\Gamma^2(\sigma)\cosh^{2\sigma}(t))$</td>
</tr>
</tbody>
</table>

Another way to analyze non-stationary signals is to expand them onto biorthogonal basis functions (referred as wavelets), constructed from shifted and scaled versions of a given mother function, $\varphi(t) \in L^2(\mathbb{R})$, keeping the energy concentrated on short intervals of $t-f$ plane. The WT spectral density, equivalent to (4.3), is performed making $t-f$ atoms (4.4) as $\varphi_{a,b}(t) = a^{-1/2}\varphi((t-b)/a)$, $a \in \mathbb{R}^+, b \in \mathbb{R}$,
then WT of \( x(t) \) defined as

\[
\tilde{\mathcal{F}}_{WT}\{x(t)\} = a^{-1/2} \int_T x(t) \varphi^* ((t - b)/a) \, dt, \quad t \in T
\]

(4.7)
is designed to procure satisfactory time resolution and poor frequency resolution at high frequencies and good frequency resolution at low frequencies. This approach makes sense, especially when the signal has high frequency components of short duration and low frequency components of long duration, which is the case in most biological signals [36]. WT has demonstrated the ability to analyze the HS more accurately than other TFR techniques like STFT or any other quadratic energy distributions in some pathological cases [36, 28]. Since WT can be expressed by means of Fourier Transform as:

\[
\tilde{\mathcal{F}}_{WT}\{x(t)\} = a^{1/2} \int_S \tilde{\mathcal{F}}_F\{x(s)\} \tilde{\mathcal{F}}_F\{\varphi^* (as)\} e^{jbs} \, ds
\]

(4.8)

with \( s \in S, a \in \mathbb{R}^+, b \in \mathbb{R} \). It can be deduced that the WT is a smoothed version of the Fourier spectrum. The spectral bandwidth of the WT can be changed, and hence, the time resolution is tuned to the speed of signal variations; being this property the most significant advantage of WT over other TFRs.

### 4.3 Parametric approach

Parametric methods of time-varying spectral analysis are based on a parametric model of signals, allowing a direct description of the dynamics which gave origin to the signal. Generally used models are linear and time invariant, known as ARMA models, whose advantages are ease of estimation and interpretability (in comparison with nonlinear models) and direct relationship with spectral information. Nevertheless these models only are adequate for stationary signals, so time varying dynamics can’t be tracked by these methods. Allowing time variation of parameters these models become more powerful and flexible. A TVARMA\((p, q)\) model, with \( p \) and \( q \) designating its autoregressive (AR) and moving average (MA) orders, respectively, is of the form:

\[
y[k] = - \sum_{i=1}^{p} a_i[k] y[k-i] \xi[k] + \sum_{i=1}^{q} b_i[k] \xi[k-i] \]

(4.9)

where \( y[k] \) the non–stationary signal to be modeled, \( \xi[k] \sim \text{WN}(0, \sigma^2_\xi[k]) \) an innovations sequence with zero mean and time-dependent variance \( \sigma^2_\xi[k] \), \( a_i \) and \( b_i \) are the model’s time-dependent AR and MA parameters, respectively.
4.4 Parameter estimation of TVAR models

We will center our attention in TVAR models, described by

\[ y[k] = - \sum_{i=1}^{p} a_i[k] y[k-i] + \xi[k] \]

\[ = a^T[k] h[k] + \xi[k] \] (4.10)

where

\[ a[k] = [a_1[k] \ldots a_p[k]]^T \]

\[ h[k] = [y[k-1] \ldots y[k-p]]^T \]

are the parameter vector and the regression vector respectively. We choose TVAR models due to their capacity to approximate most spectra, and also because estimation of ARMA models leads to non-linear estimation methods with sub-optimal or inaccurate results, whereas AR models lead to linear estimation methods which are proven to produce optimal estimates.

Anyway, it can be proven that any ARMA model can be transformed into an equivalent AR model with larger order, so, spectra modeled with ARMA models can also be modeled with AR models.

Parametric representations differ from their conventional, stationary, counterparts in that their parameters are time-dependent. The methods based upon them are known to offer a number of potential advantages listed in [98], here we recall those important for our purposes: (i) representation parsimony, as models may be potentially specified by a limited number of parameters; (ii) improved accuracy; (iii) improved resolution; (iv) improved tracking of the time-varying dynamics; (v) flexibility in analysis, as parametric methods are capable of directly capturing the underlying structural dynamics responsible for the non-stationary behavior.

Use of parametric models requires solving two problems in order to obtain the final result, which is to obtain an appropriate model for a signal. These problems are model order estimation and parameters estimation. Both problems are linked and must be solved sequentially. In the following sections we address both problems and review several methods to solve the parameter and model order estimation in TVAR models.

4.4 Parameter estimation of TVAR models

Parameter estimation in TVAR models refers to the problem of estimating the set of \( p \) time varying TVAR parameters \( \{a_i[k]\}_{i=1}^{p} \) and time varying innovations sequence power \( \sigma^2[k] \), given the set of measurements \( y[k] \). This problem is ill–posed, given that the number of parameters is larger than the number of available measurements, some assumptions should be made in order to accomplish a satisfactory estimation.
In general, it is supposed that the system changes smoothly or that the system is stationary in short segments. Under this supposition, several algorithms are devised, which we review in the following sections.

### 4.4.1 Locally stationary methods

Locally stationary methods suppose that the signal is locally stationary and regular methods such as Yule–Walker can be used to estimate parameters in short segments of length $M$ [42,92]. This method produces parameter estimates that remain constant within each segment when the segments do not overlap. In the overlapping segment case (where the active segment is forwarded by a specified forward step, say $m$), parameter estimates are obtained every $m$ samples.

For this setting, the locally stationary model would be

\[
y[k] = - \sum_{i=1}^{p} a_i y[k-i] + \xi[k] \forall k \in [k_j, k_j + M] \tag{4.11}
\]

where $[k_j, k_j + M]$ is an interval where the signal is supposed to remain stationary. This way, the parameters in this segment can be estimated as

\[
a = \arg \min_a \| y[k] + a^\top [k] h[k] \| \tag{4.12}
\]

leading to the set of Yule-Walker equations which can be solved with the Levinson-Durbin or Burg’s algorithms [100].

The critical quantity in this method is the segment length $M$. It is evident that a short length may lead to inaccurate parameter estimates, whereas a long length may not provide sufficient time resolution for adequately describing the evolution in the dynamics. Thus a compromise between achievable accuracy and time resolution is necessary. In order to achieve a reasonable accuracy, the segment length $M$ should be much larger than the model order $p$. For this reason, the method is basically suitable for cases where the evolution in the dynamics is slow. The use of overlapping segments is generally useful, yet the segment duration remains as the critical compromising factor [98].

### 4.4.2 Adaptive methods

The key idea is the formulation of an estimator of the AR parameter vector $a_i[k]$ at each time instant $k$ based on the data available until that time in a form that is recursively updated at the next time instant $k + 1$ that the next signal sample $y[k+1]$ is processed [55,80].

Most popular recursive algorithms are the least mean square (LMS) and recursive least squares (RLS) algorithms. The LMS algorithm is often considered as a
standard against other recursive algorithms, mainly because of its simplicity. The RLS algorithm is computationally more complex, but its convergence is typically an order of magnitude faster than that of LMS algorithm.

In LMS algorithm the AR parameter vector $a[k]$ is selected so that the instantaneous error between the signal $y[k]$ and the prediction $a^\top[k]h[k]$ is minimized in the mean square sense. This is:

$$
a[k] = \underset{a}{\arg\min} \ e^2[k] = \underset{a}{\arg\min} (y[k] - a^\top[k]h[k])^2
$$  (4.13)

minimization of (4.13) leads to the following set of equations [55]

$$
e[k] = y[k] - a^\top[k-1]h[k]
\quad a[k] = a[k] + 2\mu e[k]h[k]
$$  (4.14)

referred to as the LMS recursion, where the step size parameter $\mu$ controls the convergence of the algorithm. A small value of $\mu$ results in slow adaptation while a larger value gives faster adaptation but with the expense of estimate stability. The eminent feature of the LMS algorithm which has made it the most popular adaptive filtering scheme is its simplicity. The major problem of this algorithm is its slow convergence [37], which is a very important to track parameter changes of a TVAR model. There are several modifications of the basic LMS algorithm trying to improve the convergence problem, nevertheless they can’t reach an optimal performance, principally due to the approximation of the LS functional in (4.13).

In order to improve the adaptation of the LMS algorithm the approximation of the LS functional (4.13) is replaced with the weighted LS functional

$$
a[k] = \underset{a}{\arg\min} l(a[k]) = \underset{a}{\arg\min} \sum_{i=1}^{k} \lambda^{k-i}(y[i] - a^\top[i]h[i])^2
$$  (4.15)

where $\lambda$ is called the forgetting factor which must satisfy $0 < \lambda \leq 1$. The RLS algorithm is derived from (4.15) by differentiating the above functional with respect to $a[k]$. The resulting algorithm can be summarized as [55]

$$
K[k] = \frac{P[k-1]h^\top[k]}{h[h]P[k-1]h^\top[k] + \lambda}
\quad P[k] = \lambda^{-1}(I - K[k]h[k])P[k-1]
\quad a[k] = a[k-1] + K[k](y[k] - a^\top[k]h[k])
$$  (4.16)

The adaptation speed of the RLS is controlled with the forgetting factor $\lambda$. This is clearly seen from the functional (4.15). For smaller values of $\lambda$ more weight is given to the recent error terms than the first terms and, thus, the algorithm adapts faster to changes. If $\lambda = 1$ the solution returns to the regular LS solution. In practice the forgetting factor is typically chosen between $0.9 \leq \lambda \leq 1$ [116].
4.4. Parameter estimation of TVAR models

4.4.3 Smoothness priors method

Up to this point, we have addressed the problem of TVAR parameter estimation without giving any temporal evolution restriction but to remain stationary during an appropriately small interval so that the solution of the locally stationary method remain optimal, or so that the solution of the adaptive algorithms can reach balance between tracking speed and estimation effectiveness.

In this sense, giving some functional form to the evolution of the parameters can improve estimation. Starting from the fact that the real temporal evolution function of the parameters is unknown, but assuming that it smoothly varies along time, we can consider that the evolution of the parameters satisfy the following stochastic restriction [73]:

$$\nabla^n a_i[k] = \epsilon_i[k], \quad i = 1, \ldots, p$$ \hspace{1cm} (4.17)

where $\epsilon_i[k]$ is assumed to be a zero-mean white Gaussian noise sequence with variance $\sigma^2_{\epsilon_i} = \sigma^2_{\epsilon}; \quad i = 1, \ldots, p$. For the difference equation constraints $n = 1, 2, 3$,

$$k = 1 \quad a_i[k] = a_i[k - 1] + \epsilon_i[k]$$

$$k = 2 \quad a_i[k] = 2a_i[k - 1] - a_i[k - 2] + \epsilon_i[k]$$

$$k = 3 \quad a_i[k] = 3a_i[k - 1] - 3a_i[k - 2] + a_i[k - 3] + \epsilon_i[k]$$ \hspace{1cm} (4.18)

Define the $np$ component state vector $x[k]$ by

$$x[k] = [a_1[k], \ldots, a_p[k], \ldots, a_1[k - p + 1], \ldots, a_p[k - p + 1]]^T$$

Then the observations $z[1], \ldots, z[N]$ and the linear difference equation constraint models for the AR coefficients, can be expressed as the signal model in state-space representation

$$x[k] = Fx[k - 1] + Gw[k]$$

$$z[k] = H[k]x[k] + \xi[k]$$ \hspace{1cm} (4.19)

In (4.19) $H[k]$ is the $np$ vector, $H[k] = [z[k - 1], \ldots, z[k - p], 0, \ldots, 0]$ and $w[k]$ is the $p$ vector $w[k] = [\epsilon_1[k], \ldots, \epsilon_p[k]]^T$. For the difference equation orders $n = 1, 2, 3$

---

\(^1\)For a broader explanation of the smoothness priors method the reader is referred to Appendix B, or the publications of Kitagawa [75,73,72] or Poulimenos [98]
the matrices $F$ and $G$ are

\[
\begin{align*}
F &= I_p & G &= I_p \\
F &= \begin{bmatrix} 2I_p & -I_p \\ I_p & 0 \end{bmatrix} & G &= \begin{bmatrix} I_p \\ 0 \end{bmatrix} \\
F &= \begin{bmatrix} 3I_p & -3I_p & I_p \\ I_p & 0 & 0 \\ 0 & I_p & 0 \end{bmatrix} & G &= \begin{bmatrix} I_p \\ 0 \\
\end{align*}
\] (4.20)

The $np + 1$ vector $[w[k] \, \xi[k]]^\top$ is assumed to be independent with time, with

\[
[w[k] \, \xi[k]] \approx \mathcal{N} \left( \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} Q & 0 \\ 0 & \sigma^2 \end{bmatrix} \right)
\]

\[
Q = \begin{bmatrix} q_1 & 0 & \ldots & 0 \\ 0 & q_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & q_p \end{bmatrix}
\]

Given the state-space model for the smoothness priors constraints on the TVAR model, the fitting of the model to the data $z[1], \ldots, z[N]$ is achieved by maximizing the likelihood computed by the Kalman filter.

It must be underlined that although the AR coefficients in this model are expressed as the solution to a stochastic difference equation, the evolution of those AR coefficients has an equivalent formal deterministic interpretation.

### 4.4.4 Functional approximation methods

Another form to give a temporal evolution restriction to the time-varying parameters is to directly impose a functional form so that the coefficients $a[k]$ may be approximated satisfactorily by a weighted combination of a small number of known functions. The time-dependent coefficients $a[k]$ can be expressed as [48]:

\[
a_i[k] = \sum_{l=0}^{m} a_{il} f_l[k]
\] (4.21)

where the set of functions $f_l[k]$ are known as basis functions, which can be, for example, Legendre polynomials, Fourier basis, splines, etc. The election of this functions allow fast evolutions of the coefficients, but in a somewhat regular manner, in the sense that if the first derivatives of the coefficients may be arbitrarily great, higher order derivatives necessarily vanish. This provides a model for smoothed but possibly rapid evolution of the parameters. It is possible to incorporate into this framework many other functions, taking advantage of any a priori information, such as the presence of a jump in the coefficients at a known instant, seasonal effects, etc.
4.4. Parameter estimation of TVAR models

Multiplying the parameter in (4.21) with its corresponding signal value \( y[k - i] \), it is obtained

\[
a_i[k]y[k - i] = \left( \sum_{l=0}^{m} a_{il} f_l[k] \right) y[k - i] = \sum_{l=0}^{m} a_{il} (f_l[k]y[k - i])
\]

(4.22)
defining the vector \( y = [f_0[k]y[k] \ldots f_m[k]y[k]]^\top \), (4.22) becomes

\[
a_i[k]y[k - i] = [a_{i0} \ldots a_{im}]y[k - i]
\]

(4.23)

Under this parametrization a pure autoregressive process may be written as

\[
y[k] = -\sum_{i=1}^{p} a_i[k]y[k - i] + \xi[k] = -\sum_{i=1}^{p} [a_{i0} \ldots a_{im}]y[k - i] + \xi[k]
\]

\[
= -[y^\top[k - 1] \ldots y^\top[k - p]] a_{fs} + \xi[k]
\]

(4.24)
where

\[
a_{fs} = [a_{10} \ldots a_{1m} \ldots a_{p0} \ldots a_{pm}]
\]

The interpretation of (4.24) is immediate: the parameter vector \( a_{fs} \) is the vector of the regression of the nonstationary process \( y[k] \) on the \( p \) past samples of the vector \( y[k] \).

This vector process might be called a coordinate process because it is composed of the coordinates of the trajectory of \( y[k] \), when viewed as a time function, upon the basis of functions \( f_0[k] \ldots f_m[k] \). Here lies the advantage of the basis parametrization: a linear non–stationary problem becomes a linear time-invariant problem by replacing a scalar process with a vector one. The structure of the model is then defined by the parameters \( p \) the autoregressive order and \( m \) the degree of the basis. The number of unknowns is multiplied by \( m + 1 \) but this seems a small price to pay, compared to the benefit of keeping the problem linear.

In order to obtain the estimates of the parameter vector \( a_{fs} \) we minimize the functional

\[
a_{fs} = \arg \min_{a_{fs}} E\{e[k]^2\} = \arg \min_{a_{fs}} E\{y[k] + Y^\top[k] a_{fs}\}
\]

(4.25)

\[
Y[k] = [y^\top[k - 1] \ldots y^\top[k - p]]^\top
\]

leading to the least squares estimate [48]

\[
a_{fs} = (\bar{Y} \bar{Y}^\top)^{-1} \bar{Y} y
\]

(4.26)
with

\[
y = [y[1] \ldots y[N]]^\top, \quad \bar{Y} = [Y[1] \ldots Y[N]]^\top
\]
4.5. Estimation of time–dependent variance

The estimation of the innovations variance projection coefficients may be achieved by the following procedure. An initial estimate of the estimated residual series \(e[k|a_{fs}]\) variance is first obtained via a non-causal moving average filter (using a sliding time window) as follows:

\[
\sigma^2_e[k] = \frac{1}{2M + 1} \sum_{l=k-M}^{k+M} e^2[k|a_{fs}]
\] (4.27)

with \(2M + 1\) designating the window length. An initial estimate of the projection coefficient vector \(s\) may be then obtained by fitting the obtained variance \(\sigma^2_e[k]\) to a selected functional subspace. This leads to the overdetermined set of equations [98]:

\[
\sigma^2_e[k] = \sum_{l=0}^{m_s} s_l f_l[k] = f^\top[k]s
\] (4.28)

with \(f^\top[k] = [f_0[k] \ldots f_{m_s}[k]]^\top\), \(s_l\) the coefficients of the expansion and \(m_s\) the order of the expansion. Then the coefficients of the innovations variance are obtained as

\[
s = \arg\min_s (\sigma^2_e[k] - f^\top[k]s)^2
\]

obtaining the estimate

\[
s = (FF^\top)^{-1} F\sigma^2_e
\] (4.29)

with

\[
F = \begin{bmatrix} f[1] & \ldots & f[N] \end{bmatrix}, \quad \sigma^2_e = \begin{bmatrix} \sigma^2_e[1] & \ldots & \sigma^2_e[N] \end{bmatrix}^\top
\]

4.5 Estimation of time–dependent variance

The variance of the observation noise \(\sigma^2_e[t]\) is estimated, using a sliding window on the square of estimation error [116]:

\[
\hat{\sigma}^2_e[t] = \frac{1}{M} \sum_{\tau=t-M}^{t} w[\tau, \alpha]e^2[\tau]
\] (4.30)

where \(w[\tau, \alpha]\) is a smoothing Gaussian window with aperture of value \(\alpha\).

The innovations variance can also be estimated with the smoothness priors method. Consider a realization of white noise \(s[k]|k = 1, \ldots, N\) where \(s[k] \approx \mathcal{N}(0, \sigma^2[k])\) with unknown time varying variance \(\sigma^2[k]\). The stochastic process \(\chi^2[m]\) defined by [73]

\[\chi^2[m] = \frac{1}{2} (s^2[2m - 1] + s^2[2m])\] (4.31)
4.6 Model order estimation

constitutes an independent sequence of chi-square random variables with two degrees of freedom \( \chi^2[m] \sim \chi^2_2 \). Then, the transformation

\[
t[m] = \ln \chi^2[m] + e
\]

where \( e = 0.56621 \) is the Euler constant, leaves the independent random variable \( t[m] \) with distribution that is almost normal with the moments

\[
\begin{align*}
E\{t[m]\} &= \ln \sigma^2[m], \\
\text{var}\{t[m]\} &= \frac{\pi^2}{36}
\end{align*}
\]

That transformation justifies the use of a least squares type procedure for the estimation of \( t[m] \), and hence for the estimation of the unknown variance \( \sigma^2[2m] \).

To obtain a smooth estimate of the variance \( \sigma^2[m] \), consider a \( n \)-th–order difference equation constraint on the log variance defined by

\[
\nabla^n t[m] = w[m] \quad (4.33)
\]

where \( w[m] \sim \mathcal{N}(0, \tau^2) \) iid. Then, the difference equation constraint model in

\[
\begin{align*}
\mathbf{x}[k] &= F \mathbf{x}[k-1] - G w[m] \\
t[m] &= H \mathbf{x}[m] + \xi[m] \\
\begin{bmatrix} w[m] \\ \xi[m] \end{bmatrix} &\approx \mathcal{N} \left( \begin{bmatrix} 0 \\ \tau^2 0 \end{bmatrix}, \begin{bmatrix} \sigma^2 \\ 0 \sigma^2 \end{bmatrix} \right)
\end{align*}
\]

assuming that \( n = 2 \). Then the state vector is defined by \( \mathbf{x}[m] = [t[m] t[m-1]]^\top \), and the matrices \( F, G, H \) in (4.34) are

\[
F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H^\top = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

Applying the Kalman predictor and smoothing algorithms described in Section 11-C yields the smoothed value of \( t(k|N) \), the logarithm of the smoothed estimate of the changing variance. \( \sigma^2(2m|N) = \sigma^2(2m-1|N) = \exp(t(m|N)) \) is then the smoothed estimate of the changing variance.

4.6 Model order estimation

Model order selection refers to the estimation of the proper model order within a selected methodology; this is, the locally stationary, adaptive, smoothness priors or functional series methods. In the case of locally stationary, adaptive and smoothness priors methods, the order is just the TVAR model order. In the case of functional series method, one has to take into account the functional order as well.
4.7 Properties of TVAR models

Model order selection is generally based upon either trial-and-error or integer optimization schemes, according to which models corresponding to various candidate orders are estimated, and the one providing the best fitness to the non-stationary signal is selected.

The fitness function is generally taken as a tradeoff the Gaussian log-likelihood function and the number of parameters of each candidate model. The particular model that maximizes it is the most likely to be the actual underlying model responsible for the generation of the measured signal, in the sense that it maximizes the probability of having provided the measured signal values, and is thus selected.

Several fitness functions have been devised according with this principle, among them the most popular ones are the Akaike’s information criterion and Bayesian information criterion [110], both based on the Kullback–Leibler information function, which measures the quantity of shared information between two probability functions, in our case the true probability of observed data and the probability of the model for the data [77].

In general, the order selection rules are of the form

\[ IC = -2\mathcal{L}(a[k], \sigma_e^2|\{y[k]\}_{k=1}^N) + \eta(p, N)p \]  

(4.35)

where \(\mathcal{L}(a[k], \sigma_e^2|\{y[k]\}_{k=1}^N)\) is the likelihood function of the model given the measurements \(\{y[k]\}_{k=1}^N\) which penalizes the models with low matching with the measurements, and \(\eta(p, N)\) is a penalty function which discourages model overfitting, and changes depending of each information criterion:

- AIC: \(\eta(p, N) = 2\)
- BIC: \(\eta(p, N) = \ln N\)

In order to use these rules for order selection in a specific problem, the likelihood function of the model should be found.

The Gaussian log-likelihood function of a TVAR model given the signal samples \(\{y[k]\}_{k=1}^N\) is [98]:

\[ \mathcal{L}(a[k], \sigma_e^2|\{y[k]\}_{k=1}^N) = -\frac{N}{2} \ln 2\pi - \frac{1}{2} \sum_{k=1}^N \left( \ln(\sigma_e^2[k]) + \frac{e^2[k]}{\sigma_e^2[k]} \right) \]  

(4.36)

More specific likelihood functions can be obtained for each of the estimation methods studied in Sec. 4.4 (see for example [73]). Nevertheless the likelihood function in (4.36) is very general and fits with all schemes of parameter estimation.

4.7 Properties of TVAR models

Once a TVAR representation has been obtained, model-based analysis may be performed. This includes the computation of non-parameterized representations (such
as the model’s impulse response function, covariance function, and time-frequency
distributions) which are now obtained based upon the TVAR representation.

### 4.7.1 Impulse response function

As shown by Cramer [24], any non-stationary stochastic signal $x[k]$ that without
any deterministic components, such as trends, possesses a causal representation by
means of convolution of the form

$$x[k] = \sum_{l=-\infty}^{k} h[k, l] \xi[l]$$ \hspace{1cm} (4.37)

with $\xi[k]$ designating a zero mean innovations sequence and $h[k, l]$ the model’s time-
dependent impulse response function. This function is defined as the model’s re-
sponse to a discrete impulse excitation applied at time $k$. The convolution repre-
sentation may be also written as

$$x[k] = \sum_{l=0}^{\infty} h_l[k] \xi[k - l] = \sum_{l=0}^{\infty} h[k, k - l] \xi[k - l]$$ \hspace{1cm} (4.38)

Once a TVAR representation has been obtained, the impulse response function may
be computed as [49]

$$h[k, l] = \begin{cases} 
0, & k < l; \\
1 - \sum_{i=1}^{k-l} a_i[k] h[k - i, l], & k = l; \\
- \sum_{i=1}^{k-l} a_i[k] h[k - i, l], & k > l; 
\end{cases}$$ \hspace{1cm} (4.39)

with $a_i[k]$ designating the $i$-th time-dependent AR parameter.

The impulse response function $\bar{h}[k, l]$ is defined as the impulse response of the
normalized innovations variance TVAR representation:

$$x[k] = \sum_{l=-\infty}^{k} \bar{h}[k, l] \bar{\xi}[l]$$ \hspace{1cm} (4.40)

with $\bar{\xi}[k] \triangleq \xi[k]/\sigma_\xi^2[k]$ the innovations sequence normalized to unit variance, and
$\sigma_\xi^2[k]$ the innovations sequence variance. This normalization allows for the system’s
time-dependent gain $\sigma_\xi^2[k]$ to be incorporated into the impulse response function. In
this way, $\bar{h}[k, l] = h[k, l] \sigma_\xi^2[l]$. May 21, 2009
4.7.2 Covariance function of a TVAR model

The signal’s covariance function may be obtained via the normalized model’s impulse response function as follows \[98\]:

\[
\text{cov}\{x[k_1], x[k_2]\} = E\{x[k_1]x[k_2]\}
\]

\[
= E \left\{ \left( \sum_{i=-\infty}^{k_1} \bar{h}[k_1, i]\bar{\xi}[i] \right) \left( \sum_{j=-\infty}^{k_2} \bar{h}[k_2, j]\bar{\xi}[j] \right) \right\}
\]

\[
= \sum_{i=-\infty}^{\min(k_1,k_2)} \bar{h}[k_1, i]\bar{h}[k_2, i]
\]

or equivalently, by defining \(k_1 \triangleq k, k_2 \triangleq k + l\)

\[
\text{cov}\{x[k_1], x[k_2]\} = \sum_{i=-\infty}^{\min(k,k+l)} \bar{h}[k, l]\bar{h}[k + l, i]
\] (4.41)

4.8 Time–frequency distributions

The well-known, in the stationary case, notion of power spectral density is with no direct counterpart in the non-stationary case.

A notion of frequency response may be introduced in a way that is analogous to that of time-invariant systems, yet it lacks the important properties and physical significance of the latter. Indeed, the frequency response function may be defined as the system’s response to a complex exponential excitation \(e^{j\omega k}\) divided by the excitation \([59]\):

\[
H(e^{j\omega k}, t) \triangleq \text{response of the system to } e^{j\omega k}
\]

where \(j = \sqrt{-1}\) is the imaginary unit and \(\omega\) is the frequency in \(\text{rad/s}\). Using the model’s impulse response function \(h[k, k - l]\) and the convolution relationship, this may be expressed as

\[
H(e^{j\omega k}, k) = \frac{1}{e^{j\omega k}} \sum_{l=0}^{\infty} h[k, k - l]e^{j\omega(k-l)} = \sum_{l=0}^{\infty} h[k, k - l]e^{-j\omega l}
\] (4.42)

Evidently, the frequency response function is the Fourier transform of \(h[k, k - l]\) with respect to \(k\). Yet, it has been shown \([59]\) that it can be expressed as a rational function of the model parameters only in the case that the AR parameters are independent of time, while no analytic closed-form expression for its evaluation based upon the model parameters is available.
Note that in the above, the impulse response function $\bar{h}[k, l]$ of the normalized TVAR representation should be employed in order for the system’s time-dependent gain to be accounted for.

The difficulties associated with defining a notion of frequency response carry on to the definition of a power spectral density function that is valid at each time instant.

One possible approach is to employ the concept of local covariance $\text{cov}\{x[k - l], x[k + l]\}$ (4.41). The Wigner–Ville distribution is then defined as the Fourier transform of the local covariance with respect to $k$ ($k$ considered fixed), that is:

$$S_{WV}(\omega, k) = \sum_{k=-\infty}^{\infty} \text{cov}\{x[k - l], x[k + l]\} e^{j\omega k}$$

This bears a superficial resemblance to the classical definition of the power spectral density for a stationary signal, yet it may produce negative values.

Alternative power spectral density functions may be derived by using other analogies to the stationary case. The Melard–Tjøstheim power spectral density [49, 98] (also referred to as the evolutive power spectral density) is defined as

$$S_{MT}(\omega, k) = \left| \sum_{l=-\infty}^{\infty} \bar{h}[k, l] e^{j\omega k} \right|^2$$

(4.43)

Taking the magnitude squared of the normalized model’s frequency response function $\bar{H}(e^{j\omega k}, k)$ (4.42), one has:

$$\left| \bar{H}(e^{j\omega k}, k) \right|^2 = \left| \sum_{l=0}^{\infty} \bar{h}[k, k - l] e^{-j\omega l} \right|^2 = \left| \sum_{l=0}^{\infty} \bar{h}[k, k - l] e^{-j\omega(k-l)} \right|^2$$

which, by setting $k - l = l$, leads to the Melard–Tjøstheim power spectral density of (4.43). Thus, the Melard–Tjøstheim power spectral density equals the squared amplitude of the normalized model’s frequency response function, that is:

$$S_{MT}(\omega, k) = \left| \bar{H}(e^{j\omega k}, k) \right|^2$$

(4.44)

The system’s time–varying power spectral density is obtained by utilizing a sequence of frozen stationary systems to represent the non–stationary system. By analogy to the stationary case, the power spectral density is then, for each time instant, expressed as

$$S_F(\omega, k) = \frac{\left| 1 + \sum_{i=1}^{q} b_i[k] e^{-j\omega ki} \right|^2}{\left| 1 + \sum_{i=1}^{p} a_i[k] e^{-j\omega ki} \right|^2} \cdot \sigma^2[k]$$

(4.45)

Note that this would be the power spectral density of the response signal if the system were made stationary at the time instant $k$. As such, the information
conveyed is very useful, because it represents the characteristics that the system would, hypothetically, have if it became stationary with a specific configuration, corresponding to the considered time instant \( k \).

It may be shown that the system’s frequency response function (4.42) is equal to its time–varying counterpart (4.45) plus a term which may be neglected for slowly varying systems [98]. Similar relationships may be also obtained between the time–varying and the Melard-Tjøstheim power spectral densities (4.45) and (4.44). It thus follows that in the case of slowly evolving dynamics (slowly varying systems), the system’s response characteristics may be approximated using this type of analysis.

The quantity

\[
\bar{H}_F(e^{j\omega k}, k) = \frac{1 + \sum_{i=1}^{q} b_i[k]e^{-j\omega ki}}{1 + \sum_{i=1}^{p} a_i[k]e^{-j\omega ki}} \cdot \sigma_\xi[k]
\]

may be similarly interpreted as the normalized TVARMA model’s time–varying frequency response function. The function in (4.46) can be rewritten as [59]:

\[
\bar{H}_F(e^{j\omega k}, k) = \prod_{i=1}^{q} \frac{1 - z_i[k]e^{-j\omega k}}{1 - p_i[k]e^{-j\omega k}} \cdot \sigma_\xi[k]
\]

(4.47)

known as pole–zero representation, or

\[
\bar{H}_F(e^{j\omega k}, k) = \sigma_\xi[k] \cdot \sum_{i=1}^{p} \frac{c_i[k]}{1 - p_i[k]e^{-j\omega k}}
\]

(4.48)

known as partial fraction expansion. In (4.47) and (4.48), \( \{z_i[k]\}_{i=1}^{q} \) are the roots of the numerator or zeros, and \( \{p_i[k]\}_{i=1}^{p} \) are the roots or the denominator or poles, \( \{c_i[k]\}_{i=1}^{p} \) are the time–varying residues at the poles \( p_i[k] \).

The poles can be related with corresponding modes with natural frequencies and damping ratios may be computed as:

\[
\omega_i[k] = \frac{|\ln p_i[k]|}{T_s} \text{ rad/s,} \quad \zeta_i[k] = -\cos(\arg(p_i[k]))
\]

While writing (4.48), it is assumed that the rational transfer function has no repeated pole and the order of zero is less than the order of the pole. Under these conditions, a rational transfer function can be written as a weighted sum of single pole systems, where weights may vary with time.

According to the pole loci several non–stationarities can be understood. Damped sinusoids, amplitude modulated and frequency modulated signals have specific pole loci [88]. In the case of multiple pole TVAR system the combination of single pole systems can explain a larger variety of non–stationarities, as the combination of the signals previously discussed.
Chapter 5

Dimensionality reduction in TFR

The methods of parametric time–varying analysis considered in the previous chapter cover the problem of how to obtain features from signals which evidence non–stationary behavior. As a result, a set of time varying features is obtained. Examples are TVAR parameters \((a[k], \sigma^2_e)\), and time–frequency representations \(S(n, f)\), which can be referred to as matricial features. From now on, matricial features will be noted as \(X\). Figure 5.1 shows the general pose of matricial features.

![Figure 5.1: Two dimensional feature problem.](image)

Matricial features have column–wise and row–wise relationships which contain discriminant information of the modeled process. In the case of TFR these relationships are temporal and frequential. The problem of these features is their large size and large quantity of redundant data which they contain. Thus, there is a growing need for new data reduction methods that can accurately parameterize the activity

This chapter is devoted to the issue of extracting features from matricial data that can be used by conventional classifiers. The problem resides on how to reduce dimensionality, large quantity of information and redundance on TFR, keeping information contained in column–wise and row–wise relationships contained in these data. This problem can be posed mathematically as

\[ y = M\{X_k\} \]  \hspace{1cm} (5.1)

where \( X_k = [x_1, x_2, \ldots, x_N] \in X = \mathbb{R}^{N \times M} \) a matricial feature, \( y \in Y = \mathbb{R}^n \) is the feature vector of size \( n \ll MN \), and \( M : \mathbb{R}^{N \times M} \mapsto \mathbb{R}^n \) is a map from the time dependent feature space \( X \) to the reduced feature space \( Y \), which keeps most of the information in the original space (see Figure 5.2).

![Feature extraction mapping.](Figure 5.2)

Taking into account this concept, several approaches to feature extraction from matricial data have been devised. The most rudimentary approaches consist on taking measures on matricial data, such as joint moments which are supposed to summarize information contained on TFR [115]. Therefore, by keeping all the joint moments, it’s preserved all the information in the TFR. For classification task one does not need all the joint moments of the TFR but a small subset as demonstrated in [120].

Another approach consists on inference of prototypes for each class in database and then use some distance measure from each sample in the database with the prototype. The prototype can be obtained averaging all samples from the same class, by clustering techniques (in that case it would be obtained several prototypes per class), or selecting some set of matrices that best describe remaining subjects in database [101,31].

Previous methods lack of global or local information, this is, in the case of joint moments, information of entire matrix is summarized in averages that eliminate local information; in the case of prototypes, the problem consists on correct estimation of prototypes which can generalize several conditions.
Generalization and localization of information can be achieved by averaging submatrices of data matrices, an approach that will be referred to as *tiling of matrices*; or modeling matricial data as a linear combination of basis functions, approach known as *linear decomposition methods*. In the remaining part of this chapter, these methodologies will be described.

### 5.1 Feature extraction by tiling of matrices

This approach consists on computing average energy in determined regions in time–frequency plane. To do so, a grid is used, based on a predetermined partition. Each feature $Y(i, j)$ is computed as

$$Y(i, j) = \frac{1}{N_i M_j} \sum_{n_i} \sum_{m_j} X(n, m) \quad (5.2)$$

where $n_i$ is the $i$th column span and $m_j$ is the $j$th row span. For TFR each feature represents the fractional energy of the signal in a specific frequency band and time window; thus the total feature set depicts the distribution of the signal’s energy over the TF plane. Therefore, it is expected that each feature set carries sufficient information related to the nonstationary properties of the signal and thus, it can be useful for the classification process. The feature set initially is represented as an $N_t \times M_f$ matrix, where $N_t$ is the number of time windows and $M_f$ is the number of frequency subbands, and then it is reshaped into an $n = N_t M_f$ size vector. The length of the feature vector is not the same in all cases and it depends only on the time and frequency partitions. In all cases, an additional feature is used, which is the total energy of the signal. Thus, in each case the total number of features is $N_t M_f + 1$.

![Linear grid](image)

(a) Linear grid. (b) Linear grid disposition over a TFR.

Figure 5.3: Linear grid distribution

Specific tiling is defined according to some criterion about the energy disposition in the TF plane. For example, a grid can be defined dividing the time axis in equally
5.1. Feature extraction by tiling of matrices

spaced intervals or intervals defined by the information content in some parts of the TF plane and dividing the frequency axis in equally spaced intervals as shown in Fig. 5.3.

This kind of distribution assumes that information content is equally distributed along the TF plane, however in some cases this supposition is not satisfied. In this case better performance can be achieved using information distribution on matricial data. To do so, some information measure, such as variance, can be used, allowing regions in matrix with most information to have smaller intervals and better resolution as well. Inversely, regions with poor quantity of information can have larger intervals, decreasing their effect in extracted features.

An algorithm to compute a grid which adapts to information content is summarized in the following lines

1. Compute some information measure from TFR samples on database such as variance

\[
V(n, m) = \text{var}\{X(n, m)\} = E\{(X_k(n, m) - \bar{X}(n, m))\}
\]  

(5.3)

this way, a matrix with information measures \(V_{N\times M}(n, m)\) from each point in TF surface is obtained.

2. Define a variability threshold \(V_{\text{min}}\) and a maximum number of decomposition iterations \(N_d\).

3. Divide \(V\) into four equally sized submatrices \(V_1(n_1, m_1), V_2(n_1, m_2), V_3(n_2, m_1)\) and \(V_4(n_2, m_2)\) partitioning column axis in segments \(n_1\) and \(n_2\) and row axis in segments \(m_1\) and \(m_2\).
4. Compute mean variability in each submatrix $V_i(n_i, m_i)$. If mean information $V_i > V_{\text{min}}$ and $D < N_d$ repeat steps 3 and 4 with submatrix $V_i(n_i, f_i)$. Otherwise, save TF interval $n_i, m_i$.

A graphical description of this procedure is shown in Figure 5.4. Each decomposition level may produce $4^{N_d}$ TF segments, so setting an appropriate value for threshold $V_{\text{min}}$ is very important to keep the size of representation low.

5.2 Feature extraction with linear decomposition methods

Linear decomposition methods model features as a linear combination of a set of base functions multiplied by a weight value. In this way, a feature is modeled as

$$X = \sum_{i=1}^{n_{\text{pc}}} w_i \phi_i$$

(5.4)

where $w_i$ is a weight belonging to $i$–th base function $\phi_i$. In the case of vectorial features, this is, when $x_{N \times 1}$ is a vector, Eq. (5.4) becomes

$$x = \Phi w$$

(5.5)

where $\Phi_{N \times n_{\text{pc}}}$ is a matrix with base vectors on its columns and $w_{n_{\text{pc}} \times 1}$ is a vector of weights. From this definition several algorithms to compute base vectors such as PCA or PLS have been devised. Nevertheless, the considered case of matricial features this assumption is not met, and further considerations should be assumed in order to apply these methods to matricial features.

5.2.1 Principal component analysis

Principal component analysis is a multivariate statistical procedure, where the random observations are transformed into a smaller set of uncorrelated variables called principal components (PCs). Even though, the observations are considered as random variables no assumptions about the probability densities are made in PCA. PCA was first introduced in [96,58] and is equivalent to the Karhunen-Loeve transform [71,81]. When the calculated PCs are used in any form in regression analysis the term principal component regression (PCR) is used instead of PCA [66]. Some typical applications of PCA include data reduction, feature extraction, and visualization of multidimensional data.

The starting point of PCA is the derivation of the PCs themselves. Consider that we have made $M$ observations of a random vector $x \in \mathbb{R}^N$. Vectors $x$ will
span a vector space which will be at most of \( \min\{N, M\} \) dimensions. The aim in PCA is to find \( K < \min\{N, M\} \) PCs for each observation that will cover most of the variance in the observations. The first PC \( y_1 \) for observation \( x \) is obtained as a linear combination of the elements of \( x \\
\]

where \( v_1 = [v_{11}, v_{21}, \ldots, v_{N1}]^T \in \mathbb{R}^N \) is a vector of scalar weights. The task is to find the vector \( v_1 \) so that the variance of \( \theta_1 \), i.e. the variance of \( v_1^T x \), is maximized. Clearly, to achieve the maximum variance for finite \( v_1 \) some constraints must be set. The most convenient constraint is \( \|v_1\| = 1 \). The variance of the first PC is

\[
\text{var}\{y_1\} = E\{(v_1^T x - E\{v_1^T x\})^2\} = E\{v_1^T (x - E\{x\})^2\} = v_1^T E\{(x - \mu_x)(x - \mu_x)^T\} v_1 = v_1^T C_x v_1 \triangleq \lambda_1
\]

Thus, the variance of the first PC is equal to the largest eigenvalue of the covariance matrix \( C_x \) and \( v_1 \) is the corresponding eigenvector. Next, a vector \( v_2 \) that maximizes the variance of \( v_2^T x \) with constraint that \( \text{cov}\{v_2^T x, v_1^T x\} = 0 \) is looked for. Furthermore, the third PC is obtained by finding vector \( v_3 \) that maximizes the variance of \( v_3^T x \) with constraint that \( \text{cov}\{v_3^T x, v_1^T x\} = 0 \) and \( \text{cov}\{v_3^T x, v_2^T x\} = 0 \) and so on. It turns out that, the \( k \)-th PC \( k = 1, 2, \ldots, K \) is obtained by selecting \( v_k \) to be the eigenvector of \( C_x \) corresponding to \( k \)-th largest eigenvalue \( \lambda_k \) [66].

Now consider a linear transformation from original \( N \)-dimensional space \( X \) to a \( K \)-dimensional feature space \( Y \), where \( K < N \) [10]. Representation of a point \( k \) in transformed space is given by

\[
y_k = V^T x_k \quad k = 1, 2, \ldots, N_s
\]

where \( V = [v_1, v_2, \ldots, v_{n_{pc}}] \in \mathbb{R}^{N \times n_{pc}} \) is a matrix with orthonormal columns, corresponding to \( m \) eigenvectors \( w_k \) related with \( n_{pc} \) larger eigenvalues of the covariance matrix \( \Sigma_x \) defined as

\[
\Sigma_x = \sum_{k=1}^{N}(x_k - \mu)(x_k - \mu)^T
\]

where \( \mu \in \mathbb{R}^n \) is the mean TFR. This method can also be thought as an expansion in orthogonal basis known as Karhunen–Loève transform, where each vector \( x_k \) in the dataset is represented as a weighted sum of basis functions obtained as eigenvectors.
5.2. Feature extraction with linear decomposition methods

of covariance matrix. This is:

\[ x_k = \sum_{i=1}^{n_{pc}} y_k^{(i)} v_i \]  

(5.10)

where \( y_k^{(i)} \) is the \( i \)-th element of \( y_k \).

PCA on matricial data can be done by the approach of eigenfaces [122]. Consider a set of \( N_s \) training TFR surfaces \( \{X_1, X_2, \ldots, X_{N_s}\} \) that can take values in an \( N \times M \)-dimensional space. In order to apply PCA to matricial features, each surface \( X_k \) is restructured into vector \( x_k \), recasting the data as a two-dimensional matrix \( X_{rs} \) amenable to PCA. This manipulation is possible because PCA makes no assumptions about the ordering of columns for decomposition. Thus, except for the data rearranging and meaning of the PCs, the process of decomposition is the same.

In this case, each TFR in the dataset is represented as the following linear combination:

\[ X_k(n, m) = \sum_{i=1}^{n_{pc}} y_k^{(i)} V_i(n, m) \]  

(5.11)

where \( y_k^{(i)} \) is the \( i \)-th element of \( y_k \), \( V_i(n, m) \) is \( i \)-th reshaped eigenvector and \( n_{pc} \) is the number of basis functions or rank of the covariance matrix. Figure 5.5 shows the procedure of feature extraction with PCA on TFR.

![Diagram of feature extraction with PCA eigenfaces](image)

Figure 5.5: Feature extraction with PCA eigenfaces.

It was shown in [11] that this method can extract appropriately components and condition differences in time and frequency domains while effective TFR data reduction was achieved. Also, PCA provides a far more effective means of dimensionality
5.2. Feature extraction with linear decomposition methods

reduction than other feature selection methods and can effectively accommodate the loosely structured waveforms [35]. It was pointed out that applying PCA doesn’t assure separation of class distributions, and the density functions of classes may not be clearly discriminated [6]. To overcome this problem, it is possible to use some class of transformation on projected features from PCA that allows maximum separability between classes [21].

Another approach is to compute independent transformations using rows and columns of matrix as independent individuals. In this way, each surface $X_k$ is concatenated in a matrix of matrices on which is carried out PCA to obtain a transformation matrix $V$ with dimension $M \times n_{rpc}$ where $n_{rpc}$ is the number of principal components for rows [132]. This transformation will take into account row–wise relationships. In order to take into account column–wise relationships, the same process is applied on a matrix of matrices built with $X_k^\top$ to obtain a transformation matrix $W$ with dimension $N \times n_{cpc}$ where $n_{cpc}$ is the number of principal components for columns [134]. So, each matrix feature is transformed to a reduced matrix $Y \in \mathbb{R}^{n_{rpc} \times n_{cpc}}$ given by

$$Y = V X W^\top$$

This process will be referred as 2D–PCA and is depicted in Fig. 5.6. As a result, in the case of TFR, dimension reduction takes into account not only instant–by–instant variability of each random variable, but also check for information variability through the frequency spectra.
5.2. Feature extraction with linear decomposition methods

5.2.2 Partial Least Squares

The origins of PLS are traced to Herman Wold’s original non-linear iterative partial least squares (NIPALS) algorithm, an algorithm developed to linearize models which were non linear in the parameters [131]. The NIPALS method was adapted for the overdetermined regression problem a problem typically addressed with principal component regression and that extension was termed partial least squares [?].

This technique also constructs a set of linear combinations of the inputs for regression, but unlike principal components regression it uses \( y \) (in addition to \( X \)) for this construction. Like principal component regression, partial least squares (PLS) is not scale invariant, so it is assumed that each \( x_k \) is standardized to have mean 0 and variance 1. PLS begins by computing \( v_{1k} = \langle x_k, y_k \rangle \) for each \( k \). From this the derived input \( z_1 = \sum_j v_{1k} x_j \) is constructed, which is the first partial least squares direction. Hence in the construction of each \( z_m \), the inputs are weighted by the strength of their univariate effect on \( y \). The outcome \( y \) is regressed on \( z_1 \) giving coefficient \( \hat{\theta}_1 \), and then \( x_1, \ldots, x_p \) is orthogonalized with respect to \( z_1 \). This process is continued, until \( n_{pc} \leq N \) directions have been obtained [41,54]. In this manner, partial least squares produces a sequence of derived, orthogonal inputs or directions \( z_1, z_2, \ldots, z_{n_{pc}} \).

Feature extraction with the goal of achieving class separation with PLS should be superior to PCA. That is, the dimension reduction provided by PLS in a discriminant application is guided explicitly by among-groups variability, while the dimension reduction provided by PCA is guided only by total variability [8].
Chapter 6

Classification of phonocardiographic signals

Cardiac mechanical activity is appraised by auscultation and processing of heart sound records (known as phonocardiographic signals - PCG) which is an inexpensive and non-invasive procedure. Since, computer–based analysis of heart sounds may contribute to improve diagnosis of cardiac malfunctions, PCG has preserved its importance in many medical fields of clinical practice [2, 106, 34]. Specifically, systolic and diastolic murmurs, are some of the basic signs of pathological changes to be identified, but they overlap with the cardiac beat, thereby these heart sounds can not be easily separated by the human ear. Moreover, cardiac sounds are non–stationary signals and that exhibit sudden frequency changes and transients. For this reason time–frequency representations (TFR) had been proposed before to investigate the correlation between the time–frequency (t–f) characteristics of murmurs and the subjacent cardiac pathologies [106].

For that matter, both non-parametric and parametric estimations of TFR are generally employed [83, 98, 20]. Former estimation methods are commonly implemented in biomedical applications by Wavelet transform. This method, however, suffers from the same kind of tradeoff between time and frequency resolutions as the rest of the method of non–parametric estimations [116].

Since there are large differences in the transition patterns among the individual sets of signals, stable estimation of the transition pattern should be carried out. In this line, due to its intrinsic generality and its capacity to detect formant frequencies, time varying autoregressive (TVAR) models had provided useful representations of non-stationary time series in biomedical signal analysis [18]. The frequency resolution of parametric methods is superior because of the implicit extrapolation of the autocorrelation sequence. Furthermore, they can provide a higher resolution over the non–parametric estimations without the complication of the quadratic terms re-
Chapter 6. Classification of phonocardiographic signals

garded to the quadratic TFRs or the need to generate a high time-resolution scaling analysis wavelet of the scalograms. So, heart sounds can be modeled as a TVAR process with stochastically evolving parameters that adapt according to a linear dynamical system observed in additive white Gaussian noise [116].

A major motivation in this work is to generate a set of parametric time–frequency features extracted from PCG recordings, capable of detecting murmurs with higher accuracy than using static features and non–parametric TFR estimators. So, the aim of the present work is to evaluate the best set of dynamic features, estimated from TVAR models and extracted with different forms of linear decomposition methods, suitable for the classification of heart murmurs. As criteria of comparison classifier accuracy is suggested, namely, by using the well-known $k$–nn approach, that is assumed to be adequate, since it directly measures the distance from a test set item to each of the training set items immersed in Euclidean $t$–$f$ planes.

Figure 6.1 shows the proposed methodology for heart murmur detection based on TVAR modeling.

![Figure 6.1: Experimental outline](image)

The methodology consists of four parts:

**Preprocessing** where PCG signal is optimized to feature extraction process, this is, filtering to reduce undesired components, normalization and segmentation to separate individual heartbeats of PCG signals.

**Feature estimation** TVAR parameter estimation and parametric time-varying spectral analysis. In this step optimization of TVAR parameter estimators is critical in order to accomplish adequate representation of PCG signals.

**Feature extraction** where redundance and dimensionality of spectral features is reduced. Here, averaging and dimensionality reduction as well as dynamic relevance methods are used.

May 21, 2009
Classification and Validation where classifiers are trained and validated on feature spaces. Considered issues are nearest neighbors classifier tuning, assessing of extracted features and ROC curves evaluation.

This chapter is devoted to set up and evaluation of the proposed methodology for heart murmur detection based on TVAR modeling. Each of the considered steps of the methodology is assessed seeking the best configuration to improve methodology performance. Finally, obtained results are compared with state of the art results, leading to final conclusions on this matter.

6.1 Database

The database used in this work is made up of 45 de-identified adult subjects, who gave their informed consent, and underwent a medical examination. A diagnosis was carried out for each patient and the severity of the valve lesion was evaluated by cardiologists according to clinical routine. A set of 26 patients were labeled as normal, while another 19 are tagged as pathological ones with evidence of systolic or diastolic murmur, caused by valve disorders (see details in [30]). For each patient, 8 recordings of 12 s corresponding to the four traditional focuses of auscultation (mitral, tricuspid, aortic and pulmonary areas) were taken in the phase of post-expiratory and post-inspiratory apnea. Each record was obtained with the patient standing in dorsal decubitus position. The recording time was limited due to reduced endurance of apnea in patients suffering cardiac problems.

6.2 Preprocessing

After visual and audible inspection by cardiologists, one of the four signals was randomly picked up, taking into consideration that most of the times murmurs do not necessary appear in all focuses at once, unless they are very intense (which is an evidence of its harmfulness). An electronic stethoscope (WelchAllyn® Meditron model) was used to acquire the HS simultaneously with a standard 3-lead ECG (the DII derivation was used as a time reference because the QRS complex is clearly defined). Both signals were digitized at 44.1 kHz with 16-bits per sample. Tailored software was developed for recording, monitoring and editing the HS and ECG signals. As a whole, database holds 548 heart beats in total: 274 with murmurs (73 of diastolic class and 201 systolic) and 274 that are labeled as normal class.

Since the present study focuses on the $0 - 2000\,Hz$ frequency range, hence the signals are therein digitally band-pass filtered and normalized to absolute maximum of each $i$-th PCG recording $x_i = x_i / \max |x_i|$. Then, the inter-beat segmentation
of PCG is carried out to perform the analysis in each one of the individual beats. This is done with the aim of focusing the time-frequency analysis in certain regions of interest in the transformed plane, which are localized in fixed time intervals. The database used in this work, contains an ECG signal synchronized with PCG, thus detecting correctly the R peaks in ECG, it is possible to locate the beginning and end of each one of the beats in the sound signal. The R peaks in ECG can be detected employing algorithms for QRS complex detection, which have been widely developed in previous work [76]. In this case, it is used an algorithm based on linear filtering, Hilbert transform and adaptive threshold in order to find the QRS complexes and further location of R peaks [29].

6.3 TVAR model set up

6.3.1 Model order estimation

TVAR modeling is considerably dependent on the choice of structure to be used (the selection of the model order as well as the analysis window or the dynamic model for parameter evolution). Explicitly, it is commonly assumed that a major difficulty of parametric modeling, and in particular the use of AR models, is that the chosen model order has a large effect on the quality of the signal representation [83], and for pathological PCG signals it is often difficult to select a unique order value correctly [43, 57, 90].

It had been observed that a large linear predictor order (say 28) is necessary to model the PCG signal over its full frequency range [90]. Nevertheless, there are different approaches to drop off this order, by instance, the analysis can be limited to any desired frequency range such as low, medium or high frequency to get a lower predictor number.

But more accurate model order selection may be achieved based upon the minimization of a given fitness function, estimated along with recording, as explained in Sec. 4.6. In this case the Bayesian information criterion was used as fitness function, using the model likelihood function (4.36):

$$BIC(p) = \sum_{t=1}^{N} \left( \ln(\sigma_e^2[t]) + \frac{e_t^2[t]}{\sigma_e^2[t]} \right) + p \ln N$$  \hspace{1cm} (6.1)

where $N$ is the number of signal samples, $p$ the model order, $e[n]$ is the innovations sequence or estimation residuals and $\sigma_e^2[n]$ is innovations sequence time–dependent variance. In this work, values of $e[n]$ and $\sigma_e^2[n]$ are estimated by SP–TVAR method. Computed BIC curves for different cutoff frequency signal of frame blocking, arbitrary ranging the order of model $p$ from 1 up to 15, are shown in Figure 6.2(a), and May 21, 2009
the respective histograms, as well (Figure 6.2(b)). Model order to be selected is the one that minimizes mean BIC function, i.e. the elbow of curve, and it can be seen that the less is the cutoff frequency signal, the fewer order is to be selected for the model.

Figure 6.2: Estimation of TVAR model order

From the Figures 6.2, model order is taken \( p = 7 \), which likely matches the number given in [43, 57] for time varying AR modeling of heart sounds.

### 6.3.2 Estimation of TVAR parameters

Model parameters are estimated by both previously discussed approaches, namely, locally stationary TVAR (Section 4.4.1) and smoothness priors TVAR (Section 4.4.3). In case of former method, and under the assumption that the AR parameters over the succeeding frames do not vary rapidly, PCG recordings are divided into a series of successive data windows of short duration (33ms). The coefficients of the AR model are found by using the well known Burg Algorithm. Figure 6.4 shows estimates of first TVAR parameter on the shown PCG signal for different window lengths \( M \), so that effect of window length can be appraised. As increases, variance of estimates diminishes, this is, more smoothed.
In case of latter approach, the Kalman smoother is carried out using forgetting factor \( \lambda \) (0 < \( \lambda \) ≤ 1) for parameter estimation. Forgetting factor approach aids to diminish effect of old measures into current estimates, helping to match to time varying features of the signal [56]. Figure 6.4 shows estimates of first TVAR parameter on the shown PCG signal for different forgetting factors \( \lambda \).

Time–dependent variance is estimated with eq.(4.30) using as window function \( w[\tau, \alpha] \) a normalized Gaussian function with amplitude parameter \( \alpha \). Time–dependent variance estimates for some values of \( \alpha \) are shown in fig.6.5.
Tuning of parameters $\lambda_{opt}$ and $M_{opt}$ is carried out by computing the minimum mean squared error (MSE) of the reconstruction error for the signal under consideration, that is,

$$
\lambda_{opt} = \arg \min_{\lambda} \left( \sum_{k=1}^{N} y - \hat{y}(\lambda) \right)^2 = \arg \min_{\lambda} \left( \sum_{k=1}^{N} \left( y[k] - \sum_{i=1}^{p} \hat{a}_i[k,\lambda] y[k-i] \right) \right)^2
$$

(6.2)

where $p$ is the selected model order, $N$ the length of records, and $\hat{a}_i[k,\lambda]$ is the estimate of the $i$–th AR parameter on time $k$ given by estimator with parameter $\lambda$. Setting curves for the PCG database are shown in the fig.6.6(a) using values $\lambda = [0.95, 0.98, 0.99, 0.995, 0.999]$ The same procedure is applied to the case of window length $M$. Results are shown in fig.6.6(b) using values $M = [50, 70, 90, 110, 130]$. Optimal values are found as minimum MSE values on database. It can be seen that the forgetting factor may be adjusted to be $\lambda_{opt} = 0.98$; and window length adjusted to be $M_{opt} = 70$. 

Figure 6.5: Time–dependent variance estimates for different values of $\alpha$.

Figure 6.6: TVAR parameter estimator tuning
6.3. TVAR model set up

Tuning of parameter $\alpha_{opt}$ of time-dependent variance estimator is carried out selecting estimator such that variance of estimates is minimum, this is,

$$
\alpha_{opt} = \arg\min_\alpha \var\{\sigma^2[k, \alpha]\} = \arg\min_\alpha \sum_{k=1}^{N} (\sigma^2[k, \alpha] - \bar{\sigma}^2[k])^2
$$

(6.3)

where $\bar{\sigma}^2[k]$ is the mean value of the variance $\sigma_e[k]$ at instant $k$. In a similar way, we tune factor $\alpha$ of sliding window variance estimator given in (4.30), minimizing variance of $\hat{\sigma}_e^2$. Optimal value of window width $\alpha$ is found as minimum dispersion of time-dependent variance estimates. Figure 6.7 shows computation of parameter $\alpha$, and it can be seen that $\alpha$ can be tuned to $\alpha \sim 1$.

![Figure 6.7: Time-dependent variance estimator tuning](image)

6.3.3 Estimation of parametric TFR

Once TVAR parameters are computed, the parametric TFR can be computed with (4.45). One of the advantages of TVAR models is that the TFR can be computed for some specified frequency range. Analysis of PCG signals was accomplished in the range $0 - 400$ Hz in 256 points.

Representative illustrations of TFR are shown for normal (see Figure 6.8(a)) and murmur (see Figure 6.8(a)) recordings, respectively, estimated by LS-TVAR (top $t$-$f$ plane) and KF-TVAR (bottom $t$-$f$ plane) approaches. TFR-based images that are shown in Figure 6.8 pattern matrixes with dimension $N_f \times N_t$ where $N_f$ is the number of spectral components to be analyzed of PCG signal, $f = [0, 400]$ Hz; and $N_t$ is the number of discrete–time samples of each recording. This arrangement is intended to cover a broad range of time and a full range of frequencies through heart sound.
6.4 Feature extraction from TFR

6.4.1 Working area selection

Size of TFR matrixes is very high and hence reduction dimension procedure might be considered, before any processing is done. A straightforward observation of TFR in Figure 6.8 makes clear a big amount of nil content areas (that is, evidencing no informativity) of \( t-f \) surface. Therefore, it is strongly convenient to choose an adequate number of points on the \( t-f \) plane to achieve a stable decomposition. One approach for manipulating this matrix dimension is by downsampling or even decimating the time and frequency resolution of the TFR. But in this case, information content is assumed to be equally distributed along the TFR. However, since the irregular concentration of information, a better way to adjust that number is by cropping low informative areas that lie adjacent to the border of the \( t-f \) plane. A way to consider information content is to compute pointwise variance on different TFR classes

\[
V^{(i)}(t, f) = \text{var}\{S^{(i)}(t, f)\}
\]

where \( S^{(i)}(t, f) \) is a TFR corresponding to the \( i \)-th all possible classes \( \mathcal{C} = \{0, 1\} \) (normal, murmur). Thus, a coarse estimation of informative areas can be accomplished computing the union set of variances between clases:

\[
V(t, f) = \bigcup_{i \in \mathcal{C}} V^{(i)}(t, f)
\]

Defining some threshold \( \delta \) as the minimum value of variability, areas in TFR with variability lower than \( \delta \) can be removed from analysis. Figure 6.9 shows outlines depicting different complement sets of variances on dependence on value \( \delta \); the low threshold, the wider informative area.
6.4. Feature extraction from TFR

Figure 6.9: Computing relevant area of TFR. Different contours appraise the changes of threshold $\delta$.

For the present paper, as starting value of analysis, the variability threshold is tuned to be $\delta = 0.03$, and therefore the working relevant rectangle is allocated within framework described by the time interval $0 \leq t \leq 0.8$ s and frequency band $0 \leq f \leq 400$ Hz.

Additional to elimination of low variability areas, selection of relevant areas is also accomplished. Using 2D–PCA methodology, described in Chapter 5, time and frequency relevant areas are obtained as

$$ R(G) = \sum_i |\lambda_i \Phi_i| $$

where $\{\lambda_i\}$ and $\{\Phi_i\}$ are the set of eigenvalues and eigenvectors of the dataset. Figure 6.10 shows relative relevance plots for time and frequency axes. Higher values are related with higher relevance time instants of frequency points.

Figure 6.10: Relative relevance for time and frequency axes
6.4.2 Tiling

Once working area is selected, tiled-based approach that is described in section 5.1 is carried out, measuring the fractional energy on specific \( t-f \) areas. This is accomplished using the regular tiling and quadtree approaches. The first dividing the time and frequency axes into equally spaced intervals, as suggested for bio-signals in [124]; and second, based on the pixel variance estimates, quad tree partition scheme is performed upon TFR surface, thus making more dense partitions around the areas with more variability.

![Relationship between the number of divisions and segment variability](image)

Figure 6.11: Relationship between the number of divisions and segment variability

Square sizes are tuned so that variability among different squares remain similar. This is achieved by minimizing the variance of mean energy among squares,

\[
M = \arg \min_M \text{var}\{E(M)\}
\]

Tuning of tile size is achieved analyzing mean variability of each tile size when its changed in the interval \([10, 120]\). Figure 6.11 shows obtained results on dependence of size of tiles, evidencing that as the size of the tile decreases average variability diminishes at the expense of larger quantity of squares in the representation as well as increased computational cost. According to these results time–frequency grids are adjusted to be 22 frequency bins and 40 time segments.

6.4.3 Linear decomposition methods

Linear decomposition methods solve the problem of dimensionality reduction by modeling each TFR as a linear combination of basis functions. An obvious question is to find the number of latent variables needed to obtain the best generalization for the prediction of new observations. A common approach to find this number is to compute the percentage of variance explained by base vectors as the number
of vectors increase. Figure 6.12 shows results of this test using PCA and PLS approaches. In both methodologies the curve increases rapidly and over 15 to 25 components the slope decreases. Thus, the amounts of $n_{PC} = 16$ for PCA, and $n_{PC} = 20$ for PLS are found as starting values using the criterion of 90% of variance explained.

![Figure 6.12: Percentage of variance explained vs. number of vectors for TFR surfaces modeled with linear decomposition methods](image)

Linear decomposition methods compute a series of base vectors which model TFR surfaces. Each base vector codes necessary information to model each surface on database. So then, the first base vector, related with the largest eigenvalue, can be related with most relevant features in the surface; the second base vector with secondary features, and so on. Figure 6.13 shows first five base vectors computed with PCA and PLS approaches, when are multiplied pointwise by a TFR from systolic and diastolic murmurs.

It can be noticed for PCA that the base vectors are mostly related with S1 and S2 events. Larger differences among PCA components is emphasis on systole or diastole. First PCA component emphasizes diastole, whereas second and third component emphasizes systole. Other components add details on these heartbeat features. Close examination of PCA components corroborates that most of the information is concentrated around events S1 and S2, which may be explained mainly because of instantaneous power of PCG recording, but also on account of irregularity of cardiac rhythm (specially, for pathological cases). This fact becomes evident looking at the way that S2 intensity diminishes as number of components increases. At the same time, it can be noted that the influence of systole and diastole, wherever the heart murmurs are present, occurs in dissimilar way. In case of systolic murmurs Fig. 6.13(a), increasing influence can be tracked as number of components becomes greater. In the case of diastolic murmurs, their presence can be clustered into 2 types.
6.4. Feature extraction from TFR

Figure 6.13: First five base vectors multiplied by TFR
6.5 Feature extraction from TVAR parameters

Unlike TFR, TVAR parameters can’t be related with some physical quantity such as frequency, so their interpretability isn’t direct, nevertheless they offer a lower size compact representation of information contained in TFR. Data handling is simpler, but it must be taken into account that these parameters are functional vectorial data with large size to be applied to some classification algorithm directly. Then, linear decomposition methodologies applied to TFR can also be applied in this case.

In a similar form to the case of TFR, number of components of linear decomposition methods is obtained by the percentage of variance explained by base vectors as the number of vectors increase. Results are shown in Fig. 6.14. Using 90% of variance explained, PCA attains $n_{PC} = 9$ components and PLS $n_{PC} = 16$.

![Figure 6.14: Percentage of variance explained vs. number of vectors for TVAR parameters modeled with linear decomposition methods](image)

Relevance of TVAR parameters is evaluated with PCA decomposition eigenvals...
ues and eigenvectors methodology. Fig. 6.15 shows comparison of TVAR parameter relevance for different model orders. Top plots show relevance of time scale, while bottom plots show relevance of individual considered TVAR parameters (first $p$ values are time-variant AR($p$) parameters and last value is time-dependent variance). According to results, most relevant time instants are centered around S1 and S2. Time varying AR parameters have similar relevance with subtle increase in the last ones. Time-dependent variance always has the least value of relevance.

![Figure 6.15: Parameter relevance for model orders 4, 6 and 8](image)

### 6.6 Classification and validation

Classification and validation is accomplished on six time–frequency feature sets previously described, for the two considered TVAR parameter estimation methods LS–TVAR and KS–TVAR. Datasets are summarized in Table 6.1.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Spectral feature</th>
<th>Feature extraction method</th>
<th>Feature space size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS–TVAR ($M = 90, \alpha = 1$)</td>
<td>TFR</td>
<td>Linear grid</td>
<td>400($25 \times 16$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadtree</td>
<td>355</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCA-PLS</td>
<td>16 – 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2D(PCA-PLS)</td>
<td>118 – 130</td>
</tr>
<tr>
<td></td>
<td>TVAR par.</td>
<td>PCA–PLS</td>
<td>10 – 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2D(PCA–PLS)</td>
<td>44 – 50</td>
</tr>
<tr>
<td>KF–TVAR ($\lambda = 0.98, \alpha = 1$)</td>
<td>TFR</td>
<td>Linear grid</td>
<td>400($25 \times 16$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quadtree</td>
<td>355</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCA-PLS</td>
<td>16 – 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2D(PCA-PLS)</td>
<td>118 – 130</td>
</tr>
<tr>
<td></td>
<td>TVAR par.</td>
<td>PCA–PLS</td>
<td>10 – 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2D(PCA–PLS)</td>
<td>44 – 50</td>
</tr>
</tbody>
</table>

Table 6.1: Summary of considered datasets for classification of PCG signals.
As criteria of comparison, accuracy of performance is evaluated by \( k \)-nn classifier that has proven successful in many applications, regarding to straightforward comparison between Euclidean planes.

Training is carried out in accordance with the following stages:

- Number of neighbors tuning.
- Robustness of feature extraction methods.
  - Tiling (linear grid and quadtree).
  - Linear decomposition (number of components and information threshold).
- Robustness of feature estimation methods (window length, forgetting factor and variance estimator parameter \( \alpha \)).
- Feature selection by relevance analysis.

### 6.6.1 Number of neighbors tuning

The value with best classification accuracy in the analyzed feature spaces is obtained by means of variation of number of neighbors in nearest neighbors classifier. Results are shown in Fig. 6.16(a) for LS–TVAR and Fig. 6.16(b) for KF–TVAR. Average performance for the analyzed range \( k \in [1, 19] \) is sustained in a high value with low decrease as the number of neighbors increase. Along with decrease of average performance, deviation around this value decreases as the number of neighbors does. The value \( k = 3 \) is selected given that gives high average performance with low deviation. This value will be used for remaining tests.

![Figure 6.16: Number of neighbors vs. accuracy for different methods of estimation.](image-url)
6.6. Classification and validation

6.6.2 Robustness of feature extraction methods

Tiling

The considered parameters are the number of time divisions in linear grid, whereas the parameter $\varsigma_{\text{min}}$ is tuned for quadtree technique.

![Figure 6.17: Robustness assessing of tuning for tiled-TFR approach.](image)

In figure 6.17 is compared the variations on correct rate as the number of time divisions changes in linear tiling (Fig. 6.17(a)) and the information threshold increases (Fig. 6.17(b)) for LS–TVAR and KS–TVAR estimators. The number of frequency subbands remains fixed in 20 subbands. From results in Fig. 6.17(a) it’s observed that the accuracy mean and dispersion are relatively insensitive inside the considered range for linear grid. Quadtree feature extraction displays similar performance for LS–TVAR and KS–TVAR with minor average value for KS–TVAR estimator.

Linear decomposition methods

Setting up of linear decomposition feature extraction methods (PCA and PLS) is carried out. Considered tuning parameters are information threshold and number of components (base vectors), as shown in Fig. 6.18. As a first step, threshold value $\delta$ from Eq. (6.4) is changed to select an specific area within TFR (see Fig. 6.18(a)). PCA results are very sensitive to threshold change and high variability passing from some threshold to other is evidenced. As the $\delta$ increases performance tends to stabilize. On the other hand, PLS results seem to be more robust, as the performance remains almost stable. Threshold value can be chosen as $\delta = 0.07$. 

May 21, 2009
Dependence of number of components is shown in Fig. 6.18(b). Here is evidenced direct dependence of the number of components with the accuracy of classifier. Regardless of the estimation approach of model parameters, classification performance has the same asymptotic behavior starting from $8 - 10$ components for both PCA and PLS techniques. Unlike results in Fig. 6.12 the PLS components achieve better performance faster than PCA components.

Similarly, dependence of number of components in two dimensional PCA approach is assessed comparing the accuracy as the number of components increases (See Fig. 6.19). This dependency is evaluated separately by rows an columns to see the influence of each one on correct rate. For rows, high performance can be achieved with very low quantity of components ($n_{PC} = 6 - 8$), while for columns performance improves as the number of components increases until reaches a point where performance declines ($n_{PC} = 20 - 24$). In this case, performance of KF–TVAR Fig. 6.19(a) is better than performance of LS–TVAR Fig. 6.19(b).

Comparison of tuned linear decomposition feature extraction methods from TFR

May 21, 2009
is shown in Fig. 6.20. One dimensional PCA has discrete performance. Using more refined methods, performance improves, finally obtaining that with 2D–PCA the best performance is achieved. Also, best results were obtained for KF–TVAR parameter estimator.

Figure 6.20: Performance comparison between one dimensional and two dimensional linear decomposition approaches.

### 6.6.3 Robustness of feature estimation methods

Model order, forgetting factor and variance estimator parameter of considered TVAR estimation methods (LS–TVAR and KF–TVAR) are appraised. The first evaluated parameter is model order (see Fig. 6.21). Slight increase in correct rate is evidenced as the model order increases then performance remains stable from $p = 6 − 8$, with small cut in large orders in some cases.

Figure 6.21: Model order dependence on accuracy for considered TVAR estimators.

Evaluation of correct rate sensibility to LS–TVAR and KS–TVAR tuning parameters (forgetting factor and window length) is shown in Fig. 6.22. Larger sensitivity
to forgetting factor is evidenced in the case of KS–TVAR estimator. Forgetting factors with closer values to 1 have worst performance, given that these values oversmooth estimated values. Best performance is achieved in $\lambda = 0.98$. Otherwise, LS–TVAR estimator also shows some dependency on window length values, with better correct rate but larger variability for low window length estimators and variability stabilization with lower average value as the window size increases. The best results were obtained in $M = 100$ points.

![Graph](image1.png)

(a) Forgetting factor KF–TVAR  
(b) Window length LS–TVAR

Figure 6.22: Sensibility to parameter changes in final classification accuracy.

Figure 6.23 shows that classification accuracy achieves best performance when $\alpha = 1$, where mean value and variance reach best values.

![Graph](image2.png)

Figure 6.23: Sensibility of variance estimator in final classification accuracy.

### 6.6.4 Classification with TVAR parameters

TVAR parameters (time–varying AR parameters and time–dependent innovations variance) are used as features to classify PCG signals. Feature extraction is accomplished with the following linear decomposition methods: PCA, 2D–PCA and PLS.
6.6. Classification and validation

Results are shown in Fig. 6.24 using LS–TVAR and KS–TVAR estimation methods. Overall performance is slightly lower than achieved performance using parametric TFR. For a small number of neighbors it evidenced large variability, which shrinks as the number of neighbors increases. PCA and 2D–PCA approaches to feature extraction show lower efficacy than PLS.

![Graph](image1)

Figure 6.24: Classification results using TVAR parameters with linear decomposition methods.

### 6.6.5 Feature selection by relevance analysis

Given relevance measurements results in Fig. 6.10, the most relevant time instants and frequency points are extracted from TFR. Fig. 6.25(a) shows classifier performance as an increasing percentage of time instants and frequency points is taken, ordered by their relative relevance.

![Graph](image2)

Figure 6.25: Number of relevant variables taken vs. classification accuracy.

Selection of relevant variables is also accomplished for the methodology with TVAR parameters. Figure 6.25(b) shows results of classification as the number
of variables, ordered by relevance, increases (relevance based on results of Fig. 6.15(b)). As the number of relevant parameters increases performance increases as well, achieving maximum performance when the complete set of parameters is taken. This confirms the results in Fig. 6.15(b), where parameters had similar relevance.

6.7 Summary of results of tuned methodologies

Figure 6.26 shows ROC curves for best configuration of feature extraction and TVAR estimator methods using TFR surfaces. Table 6.2 summarizes performance measures of classifier trained with best configuration of feature extraction and TVAR estimator methods using TFR surfaces. ROC curves display similar behavior for LS–TVAR and KS–TVAR estimation methods and no particular order of feature extraction methods is evidenced. This is confirmed with results in Table 6.2 where homogeneous performance is evidenced. Accuracy, sensitivity and specificity for all methods lies around 98%, area under ROC curve (AUC) is settled over 99%, with slight improvement of KS–TVAR which lies over 99.5%. Nevertheless, after tuning, no particular difference of performance can be noticed among TVAR parameter estimation methods. In general, it can be seen that classifier shows some inclination to choose normal PCG signals as murmurs, demonstrated by slight inclination of ROC curves to false positives and slightly lower values of specificity than sensitivity.

Figure 6.26: ROC curves for feature extraction methodologies in TFR.

Analogous results are shown in Figure 6.26 which shows ROC curves for best configuration of feature extraction and TVAR estimator methods using TVAR parameters and Table 6.2 which summarizes performance measures of classifier trained with best configuration of feature extraction and TVAR estimator methods using TVAR parameters. Overall performance of classification with TVAR parameters
declines about 4 to 5 percentage points in PCA feature extraction method and \( \sim 1 \) points for PLS and 2D–PCA. Again, inclination to false positives is evidenced.

Figure 6.27: ROC curves for feature extraction methodologies in TVAR parameters.
6.8 Discussion and Conclusions

Proposed methodology for PCG signal classification based on features extracted from parametric TVAR modeling has given satisfying results. This methodology offers high performance with slightly higher computational cost compared with static features. Achieved accuracy reaches $99 \pm 0.05\%$ using parametric TFR and $98.08 \pm 0.08\%$ using TVAR parameters. Previous studies present classification results around $97.17\%$ using fractal features, $86.85\%$ using perceptual features, $94.35\%$ using spectral features and $96.39\%$ performing feature extraction on fractal and spectral features [2, ?, 30]. Comparison with these results show performance benefits using the proposed methodology. Several aspects concerning with the proposed methodology are discussed on the next lines:

- It was shown that TVAR parameter estimators (KF–TVAR and LS–TVAR) are robust to variations of tuning parameters. Larger sensitivity was evidenced for forgetting factor of KS–TVAR estimator, mostly because values near 1 over–smooth parameter estimates, smoothing as well information contained in these parameters. For the considered intervals of analysis both window length of LS–TVAR and width of variance estimator are unsensitive. Nevertheless, it is very important the initial choice of these tuning parameters and their coupling to the dynamics of the specific problem tackled.

- Classification accuracy remains stable as the order of TVAR model changes. This means that model order can take values lower than those suggested by BIC curves, which can be explained by the fact that most PCG signals from database are mono–component signals, which are successfully modeled with TVAR(2) model. Additional parameters serve to complement information given by first parameters or to model noise components of PCG signal. This also explains that as bandwidth of PCG signal was reduced with frame blocking, model order estimated by BIC reduces. Nevertheless, values obtained with BIC are satisfactory and can be used as initial values during set up of methodology.

- Parametric time–frequency representations are valuable tools for analysis of PCG signals. It was demonstrated that these representations yield information about how power content of the signal changes with time. Coupling of spectral dynamics with PCG signal dynamics is directly related with the particular estimation method; LS–TVAR estimator gave rough and noisy estimates, whereas KS–TVAR estimator gave smoother estimates. Nevertheless this difference was not evident in final results, because both methodologies had similar results. Although it is still unclear, this phenomenon can be attributed
to the fact that information of TFR is contained in gross features instead of details. The most notable difference between LS–TVAR and KS–TVAR is computational cost. KS–TVAR computational cost is far lower than that of LS–TVAR mostly because of recursive estimation, that allowed computation of estimates without the need for recomputing large matrices.

- By means of relevance analysis of matricial features it was possible to determine most influence areas of matrices. From TFR and TVAR parameters relevance analysis, it was found that PCG records have large relevance fluctuations with large emphasis on S1 and S2. In frequency appears a large concentration inside the range 0 – 250 Hz. Parameters shown sustained relevance. Classification results selecting most relevant areas demonstrated that performance was sustained and even better than classification with entire data.

- Feature extraction methods coupled adequately with parametric TFR and TVAR parameters of PCG signals. Once set up, all methods yield similar performance. Main difference among methods is flexibility and effectiveness in dimensionality reduction. Tiling–based methods, although much simpler to implement, are not as effective for dimensionality reduction in the case of TFR of PCG signals, because these data has large fluctuations of variability in time and frequency axes, and hence it is necessary to increase the number of divisions in each axis. Quadtree allows reduced reduced but is not able to maintain performance. On the other side, linear decomposition methods allow larger reduction of spaces, with the cost of an increased computational cost, mainly in PLS methodology. For almost every case, PLS results achieved best performance with lower quantity of components, so, once trained, this method is more recommendable than PCA. As was pointed out previously, this difference in performance is mostly attributed to use of labels when extracting components. This can be related directly with components explained by each methodology, on the one hand, PCA components explained most variant features of PCG’s TFR, which are S1 and S2, and on the other, PLS components were mostly related with discriminant features, which are murmurs, appearing on systole and diastole. 2D–PCA was an adequate improvement of PCA for matrices, and reaches similar accuracy of classifier than that of PLS with significant reduction of computational cost.

- Behavior of nearest neighbors classifier besides of high performance, directly demonstrates the structure of feature space. For all methods of feature extraction, best results were obtained for a small number of neighbors, which can be related with the fact that the border between classes is very rough. Possibly
best performance could be achieved using more refined algorithms, but given that performance is already high, this could be useless and non-sense.
Chapter 7

Detection of epileptic signs in electroencephalographic signals

Epilepsy is a brain pathology manifested with repetitive crisis (symptoms and/or positive neurological signs) that affects approximately 40 million people in the world, 10% of them suffering more than one convulsion by month. For these reasons, epilepsy represents an important health problem with evident repercussion in work and social activities on people with this condition.

Brain’s bioelectrical activity is generally stable, but can be altered in a specific brain cortex zone, or affect both brain hemispheres [9]. Epilepsy attacks can be generalized, focalized or undetermined. Generalized crisis begin with a synchronic alteration in both hemispheres, without any evidence of localized origin. Focused epileptic crisis begin in a localized zone of the brain. Epilepsy diagnostic is considered when crisis are repetitive.

Several methods of anatomic and functional representation have been developed to study the brain in non invasive form. A first class of methods records structural images (anatomic) of the brain with high spatial resolution and include computerized tomography and magnetic resonance. Other class of methods yields functional information about brain regions activated in some determined time instant. The most known functional brain mapping methods include positron emission tomography (PET) and functional magnetic resonance (fMRI), that detect changes in metabolic activity [7,15]. Time resolution of PET and fMRI is limited, given the lag on metabolic response of the brain which is in the range of a few seconds. Another method of brain function representation measures directly its electrical activity (electroencephalography EEG and magnetoencephalography MEG) and is characterized by high temporal resolution which provides information about temporal dynamics of neuronal activity at the expense of lower spatial resolution than fMRI [60]. EEG is composed of electromagnetic potentials recorded on several locations of the scalp.
Most specialists use EEG to diagnose epilepsy, being the complementary clinical test that most contributes to epilepsy diagnosis. In fact, when a crisis arises, EEG records enable establishment of zone of origin and clinical phenomenology of the event [62]. From this information, neurologist is able to make decisions with respect to treatment and planning of possible epilepsy surgery.

Presence of true inter ictal epileptiform discharges distinguishes epilepsy from other intermittent disorders. Nevertheless, up to 50% of the patients have normal EEG. Even before serial routine exams, diagnosis on these patients is normal in about 20% of obtained EEG records [87]. Sensitivity in elder patients is lower than 39% but in general, its specificity is very high, 3% of false positives in children and 0.5% in adults. From a clinical viewpoint, this information means that repetitive exams of inter ictal EEG don’t discard epilepsy, while presence of specific epileptiform alterations almost surely suggests presence of epilepsy. Additionally, normal EEG variations and artifacts can take form of epileptiform events, that must be taken into account because of interpretation errors that may lead to [17]. Ill founded epilepsy diagnosis affects negatively several social, psychologic and even economic aspects in the lifestyle of the patient and his family. For this reason, nowadays are being developed multiple tools devised to improve sensitivity in detection of epileptic activity in EEG as well as source localization, and diminish possibility of failure attributed to EEG specialist [87].

An automated seizure detection system can thus be of great interest in identifying EEG sections that contain suspicious behavior. The main difficulty with it lies in the wide variety of EEG patterns that can characterize a seizure, such as low-amplitude desynchronization, polyspike activity, rhythmic waves for a wide variety of frequencies and amplitudes, and spikes and waves [47]. In extracranial recordings, EMG, movement, and eye blink artifacts often obscure seizures. Thus, from the pattern recognition point of view, the problem is extremely complex.

Research in automated seizure detection began in the 1970s and various algorithms addressing this problem [85, 47, 108] have been presented. Methods for automatic detection of seizures may rely on the identification of various patterns such as an increase in amplitude [99], sustained rhythmic activity [46, 129], or EEG flattening [52]. Several algorithms have been developed based on spectral [89, 39, 109, 93, 97, 3, 45] or wavelet features [12, 1, 112, 111, 113, 104, 32], amplitude relative to background activity [89, 32] and spatial context [32, 123, 5, 26]. Chaotic features [86, 94, 69, 61] such as correlation dimension [79, 78], Lyapunov exponents [44], and entropy [70] have also been proposed to characterize the EEG signal. These features can then be used to classify the EEG signal using statistical methods [86, 94, 69], nearest neighbor classifiers [102], decision trees [97], ANNs [111, 44], support vector machines (SVMs) [45, 40], or adaptive neurofuzzy inference systems [104, 70] in order.
to identify the occurrence of seizures. It is crucial for seizure detection systems to result in high sensitivity, even if this results in a large number of false detections. Such systems can then be used to reduce considerably the amount of data that need to be reviewed; neurophysiologists can then easily discard false detections.

EEG signals are often quantified based on their frequency-domain characteristics. Consequently, epileptic seizures give rise to changes in certain frequencies bands. Typically the spectrum is estimated using the fast Fourier transform (FFT). A fundamental requirement in the FFT-based spectral analysis is the stationarity of the analyzed signal. In [22], it was suggested that EEG epochs shorter than 12 s may be considered stationary. Recent works have focused on the analysis of the $\delta$ (0.4–4 Hz), $\theta$ (4–8 Hz), $\alpha$ (8–12 Hz), $\beta$ (12–30 Hz) rhythms, and their relation to epilepsy. An epileptic signal is nonstationary, having time-varying frequency components. Time-frequency (TF) representations combine both time and frequency information into a single representation and have proven to be powerful tools for the analysis of nonstationary signals [14,116], and have been used for neonatal seizure detection [53,13].

Nowadays, one of the most popular TFR methods used in EEG analysis, and in biomedical applications in general, is the Wavelet transform [105,125]. This method, however, suffers from the same kind of tradeoff between time and frequency resolutions as the traditional spectrogram method. An improved time-frequency resolution can be obtained by using parametric spectral analysis methods based on time-varying linear models. A common approach is to use a time-varying autoregressive (AR) model [116]. The frequency resolution of parametric methods is superior because of the implicit extrapolation of the autocorrelation sequence [84]. For the same reason, the leakage effect of the classical spectral estimators, depending on the used windowing function, is suppressed.

This chapter is devoted to the problem of detecting epileptic signs in EEG signals using parametric TF analysis in order to extract several features from EEG segments. The method is divided into three stages. Initially, TF analysis is performed for each EEG segment and its time varying spectrum is acquired using parametric modeling. Then, features are extracted using tiling and linear decomposition approaches. Finally, extracted features are fed to a nearest neighbors classifier, which provides the final classification according to the specified number of categories. A dataset of 500 EEG segments is used, while the method is evaluated for two different classification problems, each of them addressing a different interpretation of the medical problem and thus different selection of EEGs from the whole EEG segment dataset is required for each classification problem. The obtained results reach state of the art methods’s performance.
7.1 Database

The analyzed EEG signals were recorded from 29 epilepsy patients with medically intractable focal epilepsies undergoing invasive presurgical diagnostics between 1993 and 2000 at the Department of Epileptology of the University of Bonn, Germany [4]. Database consists of five sets (noted as A-E) composed of 100 single channel EEG segments. These segments were selected and extracted from continuous multichannel EEG after visual inspection to avoid artifacts, like muscular activity or eye movements.

Datasets A and B consist of segments taken from scalp EEG records in five healthy people using electrode placement standard 10–20. Volunteers were woke up, relaxed with eyes open (A) and eyes closed (B), respectively. Datasets C, D and E were selected from presurgical diagnose EEG records. Signals from five patients were selected who had achieved complete control of epileptic episodes after dissection of one of the hippocampal formations, which was correctly diagnosed as the epileptogenic zone. Segments of set D were recorded in the epileptogenic zone, and segments of C in the hippocampal zone of the opposite side of the brain. While sets C and D only contain measured activity on inter–ictal intervals, set E only contains records with ictal activity. In this set all segments were selected from every record place exhibiting ictal activity. All EEG signals were recorded with an acquisition system of 128 channels, using average common reference. Data was digitized at 173,61 Hz with 12 bits resolution. Some typical waveforms of each set on database are shown in Figure 7.1.

Figure 7.1: Typical waveforms of database.
In considered analysis, the above-described dataset are organized to create two different classification problems and then the method was tested with all of them.

**Classification problem 1** All the EEG segments from the dataset were used and they were classified into three different classes: Z and O types of EEG segments were combined to a single class, N and F types were also combined to a single class, and type S was the third class. This set is the one closest to real medical applications including three categories; normal (i.e., types Z and O) with 200 records, seizure free (i.e., types N and F) with 200 records and seizure (i.e., type S) with 100 records.

**Classification problem 2** It has similar classes with the first, that is, normal, seizure-free and seizure, but not all the EEG segments from the dataset were employed. The normal class includes only the Z-type EEG segments (100 records), the seizure-free class the F-type EEG segments (100 records), and the seizure class the S-type (100 records).

### 7.2 TVAR model set up

#### 7.2.1 Model order estimation

Model order estimation is obtained using BIC as explained in Sec. 4.6. In this case the Bayesian information criterion was used as fitness function, using the model likelihood function \( 4.36 \). Computed BIC curve for each one of the considered datasets (A–E) for model orders from 2 to 14, are shown in Figure 7.2(a), and the respective histograms, as well (Figure 7.2(b)). Histogram in Fig. 7.2(b) shows that every set has different optimal orders, nevertheless, individual BIC curves in Fig.
7.2(a) have similar behavior. It can be concluded that an adequate TVAR model order for each dataset in EEG database is about 5 to 7. Selected TVAR model order is 6 in accordance with recommended value in [118].

7.2.2 Estimation of TVAR parameters and parametric TFR

Model parameters are estimated by recursive least squares RLS–TVAR (Section 4.4.1) and Kalman Smoother KS–TVAR (Section 4.4.3). For both estimators selection of tuning values is carried out in accordance with results in Section 6.3 for PCG signals and as suggested in [118]. Selected forgetting factor for RLS and Kalman smoother estimators is $\lambda = 0.98$. Parametric TFR are computed for frequency range $f = [0, 40]$ Hz using expression (4.45).

![TFR estimated with RLS](image1)

![TFR estimated with KS](image2)

Figure 7.3: Representative TFR sections of sets A–E.

7.3 Feature extraction from TFR

7.3.1 Selection of working area by variability and relevance analysis

Obtained TFR from EEG signals have 1000 time and 256 frequency points. This leads to a representation of 256000 points that will crash any classification algorithm. As was pointed out for PCG signals, TFR belonging to EEG signals also have large...
quantity of values with low informativity. In order to assess informativity of $t-f$ plane, pointwise variance is computed. Results in Figure 7.4 show that variability along time axis is almost constant, whereas variability along frequency axis has large changes. In frequency axis two frequency bands with high variability can be identified frequency range from 2 to 7 Hz with highest variability and frequency range from 10 to 17 Hz. The first frequency range can be associated with delta and theta activity, while the second range can be related with alpha and beta activity.

![Figure 7.4: Pointwise variability of TFR of EEG signals.](image)

Deeper understanding of $t-f$ plane relevance is achieved computing time and frequency axes relevance with 2D–PCA method (see Fig. 7.5). Figure 7.5(a) confirms deductions from Fig. 7.4 that relevance of time axis remains almost equal for all time instants. This can be explained because unlike the case of PCG signals, EEG signal do not display synchronized behavior. More important conclusions can be attained with Fig. 7.5(b) which demonstrates that relevance of frequency axis is localized in specific frequency bands. Shaded zones belong to different kinds of brain activity ($8 - 12$ Hz alpha activity, $12 - 30$ Hz beta activity, $4 - 8$ Hz theta activity and $0.4 - 4$ Hz delta activity). Most relevant zone lies on delta and theta frequency bands, while alpha and low beta activity have lower relevance.

Figure 7.6 shows how relevance is distributed for different EEG datasets. Relevance of set A is distributed on delta and theta activity, while for set B relevance concentrates on alpha activity. Sets C and D have larger relevance on low frequency components of delta activity. Set E is centralized on theta activity. From this experiment it was demonstrated that frequencies over 20 Hz are totally irrelevant from variability and relevance point of view, while all time instants are equally relevant. For this reason, analysis will be centered on frequency range from 0 – 20 Hz.
7.3. Feature extraction from TFR

7.3.2 Feature extraction with tiling

An uniform time–frequency partition is used to extract features from TFR. Among suggested partitions in [124] 3 time windows and 13 frequency subbands are used. Each feature is computed as average value in $i$–th time window and $j$–th frequency subband. Dependence of the number of neighbors in classification accuracy is shown in Fig. 7.7 using 1 to 19 nearest neighbors. Both approaches (RLS–TVAR and KS–TVAR) have best performance with low number of neighbors, as the number of neighbors increases, performance decreases along with dispersion. Best performance is achieved for KS–TVAR with 1 and 3 nearest neighbors.

Same test is performed using defined relevant area from 0 to 20 Hz. For classification test 1 performance shrinks about 1% whereas for classification test 2 per-
7.3. Feature extraction from TFR

Performance improves near 1%; in both cases dispersion of accuracy is enlarged. RLS–TVAR seems to be more sensitive to reduction of working area, because in both classification tests performance declines.

7.3.3 Feature extraction with linear decomposition methods

Linear decomposition methods are tested on complete TFR. Dependence of the number of components with classification accuracy is shown in Fig. 7.9 for PCA and PLS approaches ranging $n_{pc}$ from 2 to 100 and Fig. 7.10 for two-dimensional PCA. In the case of PCA and PLS, performance stabilizes around 60 components in PCA and 40 in PLS. Best performance is achieved with PLS components obtained from KS–TVAR estimates. In the case of two dimensional PCA, the number of components is changed in rows and columns, this is increasing the number of time base vectors and frequency base vectors from 1 to 20. Performance is quite sensitive to increase on number of frequency components (rows) while remains almost stable when number of time components (columns) increases.

May 21, 2009
Finally, dependence of number of neighbors in classification accuracy is assessed (see Figure 7.11). Again, accuracy declines as the number of neighbors increase. Best performance is achieved using 3 neighbors in PLS and 2D–PCA approaches.
7.4 Summary of results of tuned methodologies

7.4.1 Classification problem 1

Figure 7.12 shows ROC curves for best configuration of each feature extraction methodology with KS–TVAR estimator. Individual ROC curves for class 1 vs. class 2 Fig. 7.12(a), class 1 vs. class 3 Fig. 7.12(b) and class 2 vs. class 3 Fig. 7.12(c) are shown. ROC curve behavior of class 1 vs. class 2 and class 1 vs. class 3 is near ideal, while in the case of class 2 vs. class 3 performance decays. Feature extraction methods display the same ordering for ROC curves: PCA, grid, 2D–PCA and PLS, from lowest to highest. Small tendency to false positives can be seen in ROC curves in Figure , in this case tendency to classify normal EEG as interictal EEG.

![ROC curves](image)

(a) ROC class 1 vs class 2. (b) ROC class 1 vs class 3. (c) ROC class 2 vs class 3.

Figure 7.12: ROC curves for estimator KS–TVAR for classification problem 1.

Table 7.1 summarizes classifier performance measures for features extracted from KS–TVAR estimates in classification problem 1. Best performance is achieved for PLS and grid approaches. Classes 1 (normal) and 2 (interictal) are the easier ones to detect whereas class 3 (seizure) are the most difficult to detect, nevertheless, its specificity is the highest. These findings are sustained for all feature extraction methods, except for PLS whose indicators show sustained values of sensitivity, specificity and selectivity. Figure 7.13 and Table 7.2 show the same results for features extracted from RLS–TVAR. In general, observations of KS–TVAR are sustained, with overall decrease on performance measures using RLS–TVAR.
### 7.4. Summary of results of tuned methodologies

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>96.60 ± 2.67%</td>
<td>98.00 ± 2.58%</td>
<td>90.00 ± 11.55%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.50 ± 3.37%</td>
<td>98.00 ± 3.99%</td>
<td>99.75 ± 0.79%</td>
</tr>
<tr>
<td>Grid</td>
<td>Specificity</td>
<td>98.33 ± 2.36%</td>
<td>97.66 ± 3.27%</td>
<td>99.00 ± 3.16%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>97.91 ± 0.13%</td>
<td>100.00 ± 0.00%</td>
<td>99.22 ± 0.61%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.91 ± 0.13%</td>
<td>100.00 ± 0.00%</td>
<td>99.22 ± 0.61%</td>
</tr>
<tr>
<td>PCA</td>
<td>Accuracy</td>
<td>94.20 ± 2.90%</td>
<td>94.50 ± 4.97%</td>
<td>87.00 ± 11.60%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>97.50 ± 3.54%</td>
<td>95.67 ± 3.87%</td>
<td>99.50 ± 1.05%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>95.33 ± 2.81%</td>
<td>96.33 ± 3.99%</td>
<td>99.75 ± 0.79%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>93.47 ± 3.60%</td>
<td>94.96 ± 5.27%</td>
<td>97.57 ± 5.22%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.92 ± 0.15%</td>
<td>99.89 ± 0.21%</td>
<td>99.15 ± 0.73%</td>
</tr>
<tr>
<td>KS–TVAR</td>
<td>Accuracy</td>
<td>98.00 ± 2.11%</td>
<td>97.50 ± 3.54%</td>
<td>98.00 ± 4.22%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.50 ± 4.74%</td>
<td>98.67 ± 3.22%</td>
<td>99.75 ± 0.79%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>98.33 ± 1.76%</td>
<td>98.20 ± 4.25%</td>
<td>99.09 ± 2.87%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>97.62 ± 2.51%</td>
<td>99.98 ± 0.05%</td>
<td>99.82 ± 0.34%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.98 ± 0.05%</td>
<td>99.98 ± 0.05%</td>
<td>99.82 ± 0.34%</td>
</tr>
<tr>
<td>PLS</td>
<td>Accuracy</td>
<td>95.80 ± 3.05%</td>
<td>96.00 ± 3.16%</td>
<td>91.00 ± 9.94%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.00 ± 3.50%</td>
<td>97.00 ± 3.67%</td>
<td>99.75 ± 0.79%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>96.33 ± 1.89%</td>
<td>95.71 ± 5.20%</td>
<td>98.75 ± 3.95%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>94.73 ± 2.62%</td>
<td>99.93 ± 0.15%</td>
<td>99.61 ± 0.47%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.92 ± 0.12%</td>
<td>99.93 ± 0.15%</td>
<td>99.61 ± 0.47%</td>
</tr>
<tr>
<td>2D–PCA</td>
<td>Accuracy</td>
<td>98.00 ± 3.50%</td>
<td>96.00 ± 3.16%</td>
<td>91.00 ± 9.94%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.00 ± 3.50%</td>
<td>97.00 ± 3.67%</td>
<td>99.75 ± 0.79%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>96.33 ± 1.89%</td>
<td>95.71 ± 5.20%</td>
<td>98.75 ± 3.95%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>94.73 ± 2.62%</td>
<td>99.93 ± 0.15%</td>
<td>99.61 ± 0.47%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.92 ± 0.12%</td>
<td>99.93 ± 0.15%</td>
<td>99.61 ± 0.47%</td>
</tr>
</tbody>
</table>

Table 7.1: Summary of best results for classification with parametric TFR using KS estimator for classification problem 1.
7.4. Summary of results of tuned methodologies

Figure 7.13: ROC curves for estimator RLS–TVAR for classification problem 1.

![ROC curves for estimator RLS–TVAR for classification problem 1.](image)

(a) ROC class 1 vs class 2.  (b) ROC class 1 vs class 3.  (c) ROC class 2 vs class 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>Accuracy</td>
<td>97.60 ± 1.84%</td>
<td>94.00 ± 6.99%</td>
<td>98.00 ± 3.50%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>99.00 ± 2.11%</td>
<td>97.67 ± 2.74%</td>
<td>94.00 ± 6.99%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>98.59 ± 3.12%</td>
<td>96.75 ± 3.77%</td>
<td>98.18 ± 3.83%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>99.74 ± 0.12%</td>
<td>99.97 ± 0.10%</td>
<td>99.80 ± 0.33%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.74 ± 0.12%</td>
<td>99.97 ± 0.10%</td>
<td>99.80 ± 0.33%</td>
</tr>
<tr>
<td>PCA</td>
<td>Accuracy</td>
<td>93.20 ± 3.55%</td>
<td>98.00 ± 3.50%</td>
<td>94.50 ± 3.50%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>97.00 ± 3.50%</td>
<td>97.00 ± 2.92%</td>
<td>98.25 ± 2.65%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>94.00 ± 4.66%</td>
<td>96.50 ± 4.30%</td>
<td>93.92 ± 8.92%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>91.82 ± 5.67%</td>
<td>95.45 ± 4.30%</td>
<td>93.92 ± 8.92%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>98.34 ± 0.37%</td>
<td>99.85 ± 0.31%</td>
<td>99.46 ± 0.63%</td>
</tr>
<tr>
<td>RLS–TVAR</td>
<td>Accuracy</td>
<td>97.40 ± 2.12%</td>
<td>95.50 ± 5.50%</td>
<td>97.00 ± 6.75%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>99.50 ± 1.58%</td>
<td>99.00 ± 1.61%</td>
<td>98.75 ± 1.77%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>98.33 ± 3.24%</td>
<td>99.00 ± 1.61%</td>
<td>98.75 ± 1.77%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>97.74 ± 4.28%</td>
<td>98.55 ± 2.34%</td>
<td>95.61 ± 6.08%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.75 ± 0.14%</td>
<td>99.98 ± 0.04%</td>
<td>99.87 ± 0.16%</td>
</tr>
<tr>
<td>PLS</td>
<td>Accuracy</td>
<td>96.40 ± 2.07%</td>
<td>94.50 ± 3.69%</td>
<td>96.00 ± 6.99%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.50 ± 3.37%</td>
<td>98.00 ± 2.33%</td>
<td>99.00 ± 1.29%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>97.33 ± 3.06%</td>
<td>98.00 ± 2.33%</td>
<td>99.00 ± 1.29%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>96.30 ± 4.20%</td>
<td>97.05 ± 3.40%</td>
<td>96.36 ± 4.69%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.69 ± 0.06%</td>
<td>99.99 ± 0.04%</td>
<td>99.71 ± 0.54%</td>
</tr>
</tbody>
</table>

Table 7.2: Summary of best results for classification with parametric TFR using RLS estimator for classification problem 1.

May 21, 2009
7.4. Summary of results of tuned methodologies

7.4.2 Classification problem 2

(a) ROC class 1 vs class 2.  
(b) ROC class 1 vs class 3.  
(c) ROC class 2 vs class 3.

Figure 7.14: ROC curves for estimator KS–TVAR for classification problem 2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>Accuracy</td>
<td>96.33 ± 4.83%</td>
<td>98.00 ± 4.22%</td>
<td>92.00 ± 11.35%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>99.00 ± 3.16%</td>
<td>96.00 ± 5.68%</td>
<td>99.50 ± 1.58%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>99.00 ± 2.11%</td>
<td>93.30 ± 9.28%</td>
<td>97.50 ± 3.95%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>98.18 ± 3.83%</td>
<td>93.30 ± 9.28%</td>
<td>98.75 ± 3.95%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.91 ± 0.13%</td>
<td>100.00 ± 0.00%</td>
<td>99.22 ± 0.61%</td>
</tr>
<tr>
<td>PCA</td>
<td>Accuracy</td>
<td>93.33 ± 3.14%</td>
<td>95.00 ± 7.07%</td>
<td>95.00 ± 5.27%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>98.00 ± 6.32%</td>
<td>99.00 ± 2.11%</td>
<td>98.00 ± 2.58%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>96.50 ± 3.37%</td>
<td>98.09 ± 4.03%</td>
<td>96.00 ± 5.16%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>93.79 ± 5.82%</td>
<td>98.09 ± 4.03%</td>
<td>96.00 ± 5.16%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.92 ± 0.15%</td>
<td>99.89 ± 0.21%</td>
<td>99.15 ± 0.73%</td>
</tr>
<tr>
<td>KS–TVAR</td>
<td>Accuracy</td>
<td>99.33 ± 1.41%</td>
<td>100.00 ± 0.00%</td>
<td>100.00 ± 0.00%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>99.00 ± 3.16%</td>
<td>99.00 ± 3.16%</td>
<td>100.00 ± 0.00%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>100.00 ± 0.00%</td>
<td>99.50 ± 1.58%</td>
<td>99.50 ± 1.58%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>100.00 ± 0.00%</td>
<td>99.09 ± 2.87%</td>
<td>99.09 ± 2.87%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.98 ± 0.05%</td>
<td>99.98 ± 0.05%</td>
<td>99.82 ± 0.34%</td>
</tr>
<tr>
<td>PLS</td>
<td>Accuracy</td>
<td>96.33 ± 3.31%</td>
<td>93.00 ± 6.75%</td>
<td>89.00 ± 8.76%</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>99.00 ± 3.16%</td>
<td>95.50 ± 4.38%</td>
<td>98.00 ± 3.50%</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>97.50 ± 3.54%</td>
<td>95.50 ± 4.38%</td>
<td>95.98 ± 7.03%</td>
</tr>
<tr>
<td></td>
<td>Selectivity</td>
<td>95.61 ± 6.08%</td>
<td>92.00 ± 7.40%</td>
<td>95.98 ± 7.03%</td>
</tr>
<tr>
<td></td>
<td>AUC</td>
<td>99.92 ± 0.12%</td>
<td>99.93 ± 0.15%</td>
<td>99.61 ± 0.47%</td>
</tr>
</tbody>
</table>

Table 7.3: Summary of best results for classification with parametric TFR using KS estimator for classification problem 2.
7.5 Discussion and Conclusions

Proposed methodology for EEG signal classification based on features extracted from parametric TFR has given satisfying results. Best performance achieved with the proposed methodology is 98.00±2.11% for classification problem 1 and 99.33±1.41% for classification problem 2 using Kalman smoother and PLS for feature extraction. Comparison of results with other methods is shown in Table 7.5.
### Table 7.5: Comparison of the proposed methodology with other methodologies for classification of EEG signals

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear preprocessing filter, diagnostic artificial neural network (LAMSTAR) [93]</td>
<td>Z, S</td>
<td>97.2</td>
</tr>
<tr>
<td>Time–frequency domain features, recurrent neural network (RNN) [109]</td>
<td>Z, S</td>
<td>99.6</td>
</tr>
<tr>
<td>Entropy measures, adaptive neurofuzzy inference system (ANFIS) [69]</td>
<td>Z, S</td>
<td>92.22</td>
</tr>
<tr>
<td>Chaotic measures, surrogate data analysis [70]</td>
<td>Z, S</td>
<td>~ 90</td>
</tr>
<tr>
<td>Fast Fourier transform (FFT), decision tree (DT) [97]</td>
<td>Z, S</td>
<td>98.72</td>
</tr>
<tr>
<td>Discrete wavelet transform (DWT), mixture of expert model [113]</td>
<td>Z, S</td>
<td>95</td>
</tr>
<tr>
<td>Lyapunov exponents, recurrent neural network (RNN) [44]</td>
<td>Z, F, S</td>
<td>96.79</td>
</tr>
<tr>
<td>Discrete wavelet transform (DWT), adaptive neural fuzzy network (ANFN) [104]</td>
<td>Z, F, S</td>
<td>85.9</td>
</tr>
<tr>
<td>Time frequency (TF) analysis, artificial neural network (ANN) [124]</td>
<td>Z, S</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Z, O, N, F), S</td>
<td>97.73</td>
</tr>
<tr>
<td></td>
<td>Z, F, S</td>
<td>99.28</td>
</tr>
<tr>
<td></td>
<td>(Z, O), (N, F), S</td>
<td>97.72</td>
</tr>
<tr>
<td>Parametric time frequency analysis, feature extraction and relevance analysis</td>
<td>Z,F,S</td>
<td>99.33</td>
</tr>
<tr>
<td></td>
<td>(Z, O), (N, F), S</td>
<td>98.00</td>
</tr>
</tbody>
</table>

May 21, 2009
Several aspects concerning with the proposed methodology are discussed on the next lines:

- It was shown improved performance of KS–TVAR compared with RLS–TVAR. Final results show that performance with RLS–TVAR decays about 2 points compared with KS–TVAR. Comparison with other time–frequency methods shown in Table 7.5 shows that parametric time–frequency analysis achieves better performance.

- Parametric time–frequency representations are valuable tools for analysis of EEG signals. It was demonstrated that these representations yield information about how power content of the signal changes with time. Coupling of spectral dynamics with EEG signal dynamics is directly related with the particular estimation method; RLS–TVAR estimator gave rough and noisy estimates, whereas KS–TVAR estimator gave smoother estimates. In the case of EEG signals, the difference between estimators was reflected in the final accuracy of classification. RLS–TVAR and KS–TVAR estimators are both recursive, but computational cost of KS–TVAR is larger.

- By means of relevance analysis of matricial features it was possible to determine most influence areas of matrices. From TFR relevance analysis, it was found that EEG records have minor relevance fluctuations along time. In frequency appears two relevance concentrations the first from 0 to 10 Hz and the second from 10 to 20 Hz. Classification results selecting most relevant areas demonstrated that performance was sustained and even better than classification with entire data.

- Feature extraction methods coupled adequately with parametric TFR and TVAR parameters of EEG signals. There were large differences among feature extraction methods, being PLS and grid the most stable methods. PCA had the worst performance of the evaluated methods. Tiling–based methods, are more stable in the case of TFR of EEG signals, given that variability along time axis is very stable, thus letting use of lower quantity of time and frequency partitions. Linear decomposition methods still allow larger reduction of spaces, with the cost of an increased computational cost, mainly in PLS methodology. For almost every case, PLS results achieved best performance with lower quantity of components, so, once trained, this method is more recommendable than PCA. As was pointed out previously, this difference in performance is mostly attributed to use of labels when extracting components. This can be related directly with components explained by each methodology, on the one hand, PCA components explained most variant features of PCG’s
TFR, which are S1 and S2, and on the other, PLS components were mostly related with discriminant features, which are murmurs, appearing on systole and diastole. 2D–PCA was an adequate improvement of PCA for matrices, and reaches similar accuracy of classifier than that of PLS with significant reduction of computational cost.

- Nearest neighbors classifier shows high performance. Also demonstrates the structure of feature space. For all methods of feature extraction, best results were obtained with 3 to 5 neighbors, which can be related with the fact that the border between classes is very rough. Other methodologies compared in Table 7.5 classify with more refined algorithms, nevertheless, nearest neighbors classifier is capable of keeping high performance with these features.
Appendix A

Some issues on least squares estimation

A.0.1 The Gauss–Markov theorem

One important result from statistics asserts that least squares estimates of the parameter vector $\theta$ have the smallest variance among all linear unbiased estimates. We will make this precise here, and also make clear that the restriction to unbiased estimates is not necessarily a wise one. This observation will lead us to consider biased estimates from regularization theory.

From last section it was shown that the least squares estimate of the parameter vector $\theta$ is $\hat{\theta}_{LS} = (X^\top X)^{-1} X^\top y$. Let us write the parameter vector as a linear combination $\theta = \alpha^\top \beta$. This way, the least squares estimate is of the form

$$\hat{\theta}_{LS} = \alpha^\top (X^\top X)^{-1} X^\top y$$

(A.0.1)

Considering $X$ fixed, this is a linear function $c_o^\top y$ of the response vector $y$. If we assume that the linear model is correct, $\alpha^\top \beta$ is unbiased, since

$$E\{\alpha^\top \beta\} = E\{\alpha^\top (X^\top X)^{-1} X^\top y\}$$
$$= \alpha^\top (X^\top X)^{-1} X^\top X \beta$$
$$= \alpha^\top \beta$$

The Gauss–Markov theorem states that if we have any other linear estimator $\hat{\theta} = c^\top y$ that is unbiased for $\alpha^\top \beta$, that is, $E\{c^\top y\} = \alpha^\top \beta$, then [54]

$$\text{var}\{\alpha^\top \beta\} \leq \text{var}\{c^\top y\}$$

(A.0.2)

Now consider the mean squared error of an estimator $\hat{\theta}$ in estimating $\theta$

$$MSE(\hat{\theta}) = E\{(\hat{\theta} - \theta)^2\} = \text{var}\{\hat{\theta}\} + \left(E\{\hat{\theta}\} - \theta\right)^2$$
A.1. Regularization theory

The first term is the variance, while the second term is the squared bias. The Gauss–Markov theorem implies that the least squares estimator has the smallest mean squared error of all linear estimators with no bias. However, there may well exist a biased estimator with smaller mean squared error. Such an estimator would trade a little bias for a larger reduction in variance. This tradeoff can be tuned by means of regularization methods such as ridge regression.

A.1 Regularization theory

The term regularization arises from the area of ill-posed problems [51]. For such problems a unique solution does not exist or the solution is unstable. In the latter case, small errors in the observations can cause large error in the solution. Methods that are used to stabilize the problem such that the solution becomes unique or less sensitive to observation errors are called regularization methods. Probably the most popular such method is the Tikhonov regularization [119]. In the standard Tikhonov regularization, the functional to be minimized is of the form

\[
l(\theta) = \|y - X\theta\|_2^2 + \kappa^2 \|\theta\|_2^2 \tag{A.1.3}
\]

where \(\kappa > 0\) is a regularization parameter that controls the significance of the second term in the functional. Clearly, the objective is not just to find a minimizer of the residual norm (first term) but to allow small deviation from it in order to find a solution with smaller norm.

In the more general form of Tikhonov regularization, the functional to be minimized is

\[
l(\theta) = \|L_1(y - X\theta)\|_2^2 + \kappa^2 \|L_2(\theta - \theta^*)\|_2^2 \tag{A.1.4}
\]

where \(L_1\) is a weight matrix, \(L_2\) is a regularization matrix, and \(\theta^*\) is a prior guess for the solution. The regularization matrix \(L_2\) is typically set to be either identity matrix or a discrete approximation \(D_d\) of \(d\)th-order derivative. Methods using difference approximations in regularization can, in general, be called smoothness priors methods [73]. The difference approximations are banded matrices with full row ranks. For example, the second-order difference matrix \(D_2\) is of the form

\[
D_2 = \begin{bmatrix}
1 & -2 & 1 & 0 & \cdots & 0 \\
0 & 1 & -2 & 1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 1 & -2 & 1
\end{bmatrix} \in \mathbb{R}^{(N-2) \times N}
\]

May 21, 2009
Appendix B

Smoothness Priors Method for Function Estimation

B.1 The smoothing problem

The origin of the smoothing problem can be attributed to Whitaker [?]. Consider:

\[ y[k] = f[k] + \xi[k] \]  \hspace{1cm} (B.1.1)

with the \( \xi[k] \approx \mathcal{N}(0, \sigma^2) \), \( \sigma^2 \) unknown, and \( f[n] \) an unknown smooth function. The problem is to estimate \( \{f[k], k = 1, \ldots, N\} \) in some statistically satisfactory manner.

It was suggested that the solution \( \{f[k], k = 1, \ldots, N\} \) should balance a trade-off between infidelity to the data and infidelity to a \( n \)th-order difference equation constraint [130]. For a fixed value of \( n \) and \( \lambda \) the solution satisfies:

\[
\min_f \left\{ \sum_{k=1}^{N} (y[k] - f[k])^2 + \lambda^2 \sum_{k=1}^{N} (\nabla^n f[k])^2 \right\} \hspace{1cm} (B.1.2)
\]

The fist term in (B.1.2) is the infidelity to the data measure, the second is the infidelity to the smoothness constraint. The choice of \( \lambda \), the smoothness tradeoff parameter, is chosen according with the particular application [73]. Also in (B.1.2) the difference equation constraints are \( \nabla f[k] = f[k] - f[k-1] \), \( \nabla^2 f[k] = f[k] - 2f[k-1] + f[k-2] \), etc.

The properties of the solution are clear. If \( \lambda = 0 \), \( f[k] = y[k] \), and the solution is a replica of the observations. The sum of squares of errors is zero and the solution is uninteresting. As \( \lambda \) becomes increasingly large, the smoothness constraint dominates and the solution satisfies a \( n \)th-order least squares constraint. (For \( n = 1 \), the solution is the mean of the data, etc., for increasing \( n \).) For known \( n \) and \( \lambda \), \( f \) is a solution to a least squares problem.
Generally, for $\lambda$, $n$ fixed, the solution for $f$ can be obtained as the solution of

$$
\min_f \left\| \begin{bmatrix} x \\
0 \end{bmatrix} - \begin{bmatrix} I \\
\lambda D_n \end{bmatrix} f \right\|^2
$$

(B.1.3)

The solution is

$$
f = (I + \lambda^2 D_n^T D_n)^{-1} y
$$

(B.1.4)

with

$$
SSE(\lambda, n) = y^T y - f^T (I + \lambda^2 D_n^T D_n) f
$$

(B.1.5)

$$
\sigma^2 = \frac{SSE(\lambda, n)}{N}
$$

In (B.1.3) $D_n$, an $N \times N$ matrix, expresses the difference equation constraints on the solution $f$, $I$ is the $N \times N$ identity matrix and $SSE(\lambda, n)$ is the sum of squares of the residuals.

### B.2 The Shiller–Akaike Smoothing Priors Solution

The development below follows Shiller [7] and Akaike [8].

$$
\exp \left( -\frac{1}{2\sigma^2} \sum_{k=1}^N (y[k] - f[k])^2 \right) \cdot \exp \left( -\frac{\lambda^2}{2\sigma^2} \sum_{k=1}^N (\nabla^n f[k])^2 \right)
$$

(B.2.6)

This is proportional to

$$
p(f|y, \lambda, \sigma) \propto p(y|f, \sigma) p(f|\lambda, \sigma)
$$

(B.2.7)

Equation (B.2.7) is an interpretation of (B.2.6). To within a multiplicative constant, (B.2.7) can be identified as the posterior distribution of the solution $f$ given the parameters $\lambda$ and $\sigma$ and the data $y[1], \ldots, y[N]$. In (B.2.7) $p(y|f, \sigma)$ is the conditional data distribution and $p(f|\lambda, \sigma)$ is the prior distribution on $f$. From (B.2.6), (B.2.7) the Whittaker smoothness tradeoff parameter $\lambda^2/\sigma^2$ is therefore a signal-to-noise measure. In the Bayesian literature, $\lambda$ is referred to as a hyperparameter [9].

The integrated likelihood for $\lambda$, $\sigma^2$ and $n$ is given by

$$
L(\lambda, \sigma^2, n) = \int p(y|f, \sigma) p(f|\lambda, \sigma) df
$$

(B.2.8)

Taking minus two times the logarithm of the likelihood (B.2.8) yields the explicit closed form expression for $-2 \ln L(\lambda, n)$

$$
-2 \ln L(\lambda, n) = N \ln \frac{1}{N} SSE(\lambda, k) + \ln |I + \lambda^2 D_n^T D_n| - \ln |\lambda^2 D_n^T D_n|,
$$

(B.2.9)

The solution, which minimizes (B.2.9) may be achieved by a discrete two parameter search over the parameters $\lambda$ and $n$. In (B.2.9) $|A|$ is the determinant of the matrix $A$ and $SSE(\lambda, n)$ is as defined in (B.1.5).

May 21, 2009
B.3 State space modeling

Imbedding the smoothness priors difference constraints into a state-space form signal model and invoking the Kalman filter methodology to compute likelihoods of the hyperparameters [73], the developed smoothness-priors solution reduces from a computational complexity $O(N^3)$ to $O(N)$.

For example, the second-order difference constraint model can be expressed in the state-space form

$$x[k] = Fx[k-1] + Gw[k]$$
$$y[k] = Hx[k] + \xi[k]$$

with the state vector $x[k]$, and $F$, $G$, $H$ matrices given by

$$x[k] = \begin{bmatrix} f[k] \\ f[k-1] \end{bmatrix}, \quad F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix},$$
$$G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H^\top = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The vector $w[k]$ and $\xi[k]$ are assumed to be an uncorrelated sequence with

$$w[k] \approx \mathcal{N}(0, Q), \quad \xi[k] \approx \mathcal{N}(0, \sigma^2)$$

The signal model in (B.3.10) can be generalized to include difference equation constraints of different orders as well as a variety of difference equation constraints for trend and seasonalities.

In this algorithm, the initial values $x[0|0]$ and $P(0|0)$ must be given. For a stationary system the theoretical mean value and the covariance matrix of the state vector can be used. For a nonstationary system the theoretical mean and covariance cannot be defined. We use $x[0|0] = 0$ and $P(0|0)$ a diagonal matrix with large diagonal values. This is equivalent to estimating the initial values from the entire data set.

The joint density of the observations is

$$p(y[1], \ldots, y[N]) = \prod_{k=1}^{N} p(y[k]|y[1], \ldots, y[k-1])$$

with

$$p(y[k]|y[1], \ldots, y[k-1]) = \frac{1}{2\pi r[k]} \exp \left( -\frac{v[k]^2}{2r[k]} \right)$$

that yields the log likelihood

$$\ln L(\lambda, n) = -\frac{N}{2} \ln 2\pi - \frac{1}{2} \sum_{k=1}^{N} \ln r[k] - \sum_{k=1}^{N} \frac{v[k]^2}{2r[k]}$$
In (B.3.13) and (B.3.14)

\[ v[k] = y[k] - H[k]x[k|k - 1] \]
\[ r[k] = H[k]V[k|k - 1]H[k]^T + \sigma^2 \]

where \( v[k] \) and \( r[k] \) are, respectively, the innovations and the observation variance conditioned on the observations up to time \( k - 1 \).

The maximum likelihood estimate of the parameters of the model are obtained by maximizing (B.3.14) with respect to those parameters. The value of the AIC criterion for the fitted model is [110]

\[ AIC = -2 \ln(\text{maximized likelihood}) + 2(\text{number of fitted parameters}) \]

(B.3.15)

The minimum AIC model is defined by those parameter values that minimize the AIC. The number of fitted parameters in state-space modeling is the dimension of the state transition matrix \( F \) plus the number of estimated parameters.
Bibliography


May 21, 2009


May 21, 2009
Bibliography


Bibliography


May 21, 2009
Bibliography


