University of Kuopio Department of Applied Physics Report Series ISSN 0788-4672

Estimation theoretical background of root tracking algorithms with applications to EEG

Pasi A. Karjalainen

June 1996

Report No. 5/96

This work has been approved as a Phil.Lic. thesis in the Faculty of Natural and Environmental Sciences, University of Kuopio, June 1996

University of Kuopio • Department of Applied Physics
P.O.Box 1627, FIN-70211 Kuopio, Finland

Abstract

UNIVERSITY OF KUOPIO, Faculty of Natural and Environmental Sciences
Department of Applied Physics
KARJALAINEN PASI A.: Estimation theoretical background of root tracking algorithms with applications to EEG
Phil.lic Thesis, 73 pages
Supervisors:
Prof., Ph.D. Lauri Patomäki
Ph.D. Ari Pääkkönen

June 1996

Keywords: recursive Bayesian estimation, adaptive algorithms, time-varying models, root tracking, EEG, event related synchronization

Adaptive methods are commonly used for tracking of the time-varying characteristics of non-stationary signals. In many cases of particular interest is the tracking of spectral properties of a narrow band random signal. When using linear models the parameters of the underlying model have seldom any special meaning. With this kind of models however the roots of the model polynomials have straigth connection to spectral components of the signal. The calculation of all the roots during the iteration is usually too time consuming. In such cases a method of adaptive root tracking can be effective and usefull.

In this work some root tracking algorithms are presented. After survey of the general aspects of estimation methods and some discussion of the optimality of the estimators, the recursive approach to mean square estimation is presented. This approach is used to derive the most common adaptive algorithms used in analysis of the time-varying signals. The implicit assumptions of the algorithms are then derived to see, when they are optimal in mean square sense.

After presentation of the root tracking algoritms the performance of some of them are evaluated with simulations. Finally one of the algorithms is used for detection of the event related synchronization (ERS) changes in human EEG.

As a result of the evaluation it can be concluded that the adaptive root tracking is one possibility for tracking the time-varying spectral properties of the signal, especially time-varying narrow band signals. The tracking capability of some of the methods is the same as the capability of the underlying adaptive algorithm, and thus the critical point is the selection of the time series model and the adaptive algorithm. Adaptive root tracking can be used for tracking the properties of the EEG-signal. This is the case at least with the ERS test. However the inference about the underlying neural processes is not so straightforward. In some cases the dynamical (neural) system operates in such a mode, that the resulting activity is narrow band. In such cases, the state of the output activity of the dynamical system can fluctuate in time without the system state changes. In this kind of cases all the methods dealing with the power of the spectral components or the signal morphology, including the human investigation, can give uncorrect results.

Acknowledgements

This work was done in the Department of Applied Physics, University of Kuopio, during 1995-1996. I thank my supervisors Prof. Lauri Patomäki Ph.D. and Ari Pääkkönen Ph.D. for their support to this work. I want to thank the reviewers Docent Jouko Tervo, Ph.D. and Jari Kaipio, Ph.D. for their many, many suggestions and advices how to get my work better. Especially I want to thank Dr. Kaipio for his contributions to many results of this work.

Kuopio, June 1996

Pasi Karjalainen

Probability

$x \sim \mathcal{N}(\eta_x, C_x)$	Jointly Gaussian random vector $x = (x_1, \ldots, x_n)^T$
$p(x)$	Joint density function of x
$p(x\|y)$	Conditional density function of x given y
$\eta_x, E\{x\}$	Expected value of x
C_x	Covariance of x
R_x	Correlation of x
$\eta_{x\|y}, E\{x\|y\}$	Expected value of x given y, conditional mean
$C_{x\|y}$	Conditional covariance of x given y
$S(\omega)$	Power spectrum

Estimation

z	Vector of observations $x = (z_1, \ldots, z_n)^T$
h	Model for observation
H	Linear model for observation
θ	Parameter vector
$\hat{\theta}(z)$	Estimator of parameter vector θ based on observations z
$\hat{\theta}$	Estimate of parameter vector θ
$\tilde{\theta}$	Estimation error
$C(\theta, \hat{\theta})$	Cost function
$B(\hat{\theta})$	Bayes cost
$B(\hat{\theta}\|z)$	Conditional Bayes cost

$\hat{\theta}_{\mathrm{ML}}$ Maximum likelihood estimate

$\hat{\theta}_{\mathrm{B}}$ Bayesian estimate

$\hat{\theta}_{\mathrm{MS}}$ Mean square estimate

$\hat{\theta}_{\mathrm{LMS}}$ Linear mean square estimate

$\hat{\theta}_{\mathrm{GMS}}$ Generalized mean square estimate

$\hat{\theta}_{\mathrm{CM}}$ Conditional mean estimate

$\hat{\theta}_{\mathrm{MV}}$ Minimum variance estimate

$\hat{\theta}_{\mathrm{UC}}$ Uniform cost estimate

$\hat{\theta}_{\mathrm{MAP}}$ Maximum a posteriori estimate

$\hat{\theta}_{\mathrm{GM}}$ Gauss-Markov estimate

$\hat{\theta}_{\mathrm{LS}}$ Least squares estimate

$\hat{\theta}_{\mathrm{GLS}}$ Generalized least squares estimate

$\hat{\theta}_{\mathrm{LMV}}$ Linear minimum variance estimate

W Weighting matrix

l_{GLS} Generalized least squares index

J Jacobian

K_t Kalman gain vector

Time series

y_t Process, time series

$E\{y_t\}$ Mean of the process

$\hat{y}_{t|\theta}$ Estimated (predicted) process with parameters θ

e_t Noise process

φ_t Regressor vector

θ_t Time-varying parameter vector

$\nabla_\theta(\hat{y}_{t|\theta})$ Gradient of $\hat{y}_{t|\theta}$

ε_t Prediction error process

$\beta(t,k)$ Weighting sequence

$V_t(\theta)$ Weighted least squares index

P_t Recursive estimate for covariance or its inverse

AR Auto regressive

ARMA	Auto regressive moving average
MA	Moving average
RLS	Recursive least squares
NLMS	Normalized least mean squares
LMS	Least mean squares

Root tracking

q^{-1}	Delay operator
$A(q)$	System polynomial
ζ	Root of a polynomial
$\zeta(a)$	Coefficients to roots mapping
$\Phi(\zeta)$	Roots to coefficients mapping

CHAPTER I

Introduction

Signals are often presented as output of dynamical systems. The properties of the signal are then determined by the properties of the underlying dynamical system. If the statistical properties of the signal are for example time-invariant, the signal can often be modeled as an output of a linear time-invariant system with some input. With time-invariant systems we refer to such systems whos parameters are constant of time. The dynamical system then serves as a model for the signal and the estimation of the parameters of the dynamical system can be seen as modeling of the signal.

The usual way to construct such a model is to let it be a linear function which maps the past signal values to estimate of present signal value. The parameters of model describe the whole signal. The usual criterion used in modeling is so called mean square criterion. It says that the model parameters must be selected so that the mean of the squares of the estimation error is in minimum. We will show that this criterion is optimum among wide range of models and assumptions about data.

When the characteristics of the signal vary in time no estimator having fixed parameters is applicable. Then we have to use systems with time-varying parameters for modeling of signals. If we still use the mean square criterion the resulting algorithm which minimizes the criterion recursively is so called Kalman filter.

For using Kalman filter we have to assume some model for parameter fluctuations. The simpliest such a model would be that the parameters obey a random walk. Further we show that under certain assumptions about the parameters, the classical recursive least squares algorithm gives the same result as the optimal Kalman filter.

The linear filters or systems used in this work are in fact difference equations which can also be presented in complex rational polynomial form. In this form the coefficients of the nominator and the denominator of so called transfer function are the parameters of the time-varying signal. In some cases, for example when the signal contains narrow band components, the roots of such a polynomial are of the greater interest than the polynomial coefficients themselves. When the coefficients are time-varying the solving of all the roots numerically at each time instant would be very time consuming and unefficient. In this work we have studied some methods with which the roots of this kind of polynomials having time-varying coefficients can be tracked recursively. One of the methods is novel and is presented in [51]. Some of the methods are compared with simulations and one of the methods is used for detecting components of real EEG data.

1.1 Probability theory

We shall not review fundamental definitions of probability theory, such as probability space, elementary events, random variables and probability measure here. These are presented e.g. in [50]. For the definitions of the distribution and density functions of multidimensional random variables we refer to [55]. In this text random variables are not notated systematically differently from deterministic variables. Usually all the random variables are vector valued. The joint density of the components of the random vector $x = (x_1, \ldots, x_n)^T$ is notated with $p(x)$. The joint density of the components of two random vectors x and y are noteted by $p(x, y)$.

The mean or expected value $\eta_x \in \mathbb{R}^n$ of x is

$$\begin{aligned} \eta_x &= E\{x\} = \int_{-\infty}^{\infty} xp(x)\,dx && (1.1)\\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_1,\ldots,x_n)^T p(x_1,\ldots,x_n)\,dx_1 \cdots\ dx_n && (1.2)\\ &= (\eta_{x_1},\ldots,\eta_{x_n})^T && (1.3) \end{aligned}$$

The correlation matrix of random vector x is defined to be

$$R_x = E\left\{xx^T\right\} = \begin{pmatrix} E\{x_1x_1\} & \cdots & E\{x_1x_n\} \\ \vdots & \ddots & \vdots \\ E\{x_nx_1\} & \cdots & E\{x_nx_n\} \end{pmatrix} \quad (1.4)$$

that is, the componentwise expectation of the outer product of x with itself. The cross correlation of random vectors x and y is defined to be

$$R_{xy} = E\left\{xy^T\right\} \quad (1.5)$$

Covariance is the correlation of the random vector $(x - \eta_x)$

$$C_x = E\left\{(x-\eta_x)(x-\eta_x)^T\right\} = E\left\{xx^T\right\} - \eta_x\eta_x^T \quad (1.6)$$

and the cross covariance of x and y is defined

$$C_{xy} = E\left\{(x-\eta_x)(y-\eta_y)^T\right\} = E\left\{xy^T\right\} - \eta_x\eta_y^T \quad (1.7)$$

Clearly

$$C_{xy} = C_{yx}^T \quad (1.8)$$

The conditional density of x given y is defined to be

$$p(x|y) = \frac{p(x,y)}{p(y)} \quad (1.9)$$

whenever $0 < p(y) < \infty$ and 0 otherwise. Clearly we can also write

$$p(y|x) = \frac{p(y,x)}{p(x)} \quad (1.10)$$

and equating $p(x,y) = p(y,x)$ gives

$$p(x|y)p(y) = p(y|x)p(x) \quad (1.11)$$

This is called the Bayes theorem. The conditional mean of x given y is

$$\eta_{x|y} = E\{x|y\} = \int_{-\infty}^{\infty} xp(x|y)\,dx \quad (1.12)$$

which is a function of the random variable y. The conditional covariance of x given y is then [50]

$$C_{x|y} = E\left\{(x-\eta_{x|y})(x-\eta_{x|y})^T\middle|y\right\} = E\left\{xx^T|y\right\} - \eta_{x|y}\eta_{x|y}^T \quad (1.13)$$

Random variables x_i are said to be statistically independent if

$$p(x) = \prod_i p_i(x_i) \quad (1.14)$$

Random variables x_i are said to be jointly Gaussian if their joint density can be written in form

$$p(x) = \frac{(\det C_x)^{-1/2}}{(2\pi)^{n/2}}\ e^{-\frac{1}{2}(x-\eta)^T C_x^{-1}(x-\eta)} \quad (1.15)$$

where $\det C_x$ denotes the determinant of C_x. When x is jointly Gaussian with mean η and covariance C_x, this is denoted by

$$x \sim \mathcal{N}(\eta, C_x) \quad (1.16)$$

1.2 Stochastic processes

A scalar valued random (or stochastic) process x_t is a set $x_t : t \in T$ of random variables defined in the same probability space [47]. T is said to be the parameter set of process. When T is a finite or countably infinite set, we say that the process is a discrete parameter set process. When the parameter t denotes the time, the process is said to be a discrete time process, a time series.

The mean of the random processes x_t is a function of time

$$\eta_{x_t} = E\{x_t\} = \int_{-\infty}^{\infty} x_t p(x_t)\, dx_t \tag{1.17}$$

and covariance is a function of two time variables t and s

$$C_{x_t,x_s} = E\left\{(x_t - \eta_{x_t})(x_s - \eta_{x_s})^T\right\} = E\left\{x_t x_s^T\right\} - \eta_{x_t}\eta_{x_s}^T = R_{x_t,x_s} - \eta_{x_t}\eta_{x_s}^T \tag{1.18}$$

If the mean is time-invariant

$$E\{x_t\} = \eta \tag{1.19}$$

we say that the process is first order stationary. If in addition the autocorrelation R_{x_t,x_s} depends only on the time difference $\tau = s - t$

$$R_{x_t,x_s} = R_\tau \tag{1.20}$$

we say, that x_t is second order stationary. The term wide sense stationarity is also used for second order stationarity. We abbreviate this as WSS.

Special process is a white noise process. Discrete white noise process is a sequence of independent zero mean random variables [60].

Suppose that x_t is a WSS process with autocorrelation R_τ. Then its power spectrum is defined as

$$S(\omega) = \sum_{\tau=-\infty}^{\infty} R_\tau e^{-j\tau\omega} \tag{1.21}$$

The spectrum is thus the discrete Fourier transform of the autocorrelation of the x_t.

1.2.1 Models for time series

Some classes of discrete stochastic processes can be expressed as outputs of linear time-invariant systems with white noise input. We call the process y_t autoregressive moving average process or ARMA(p, q)-process if it can be expressed as

$$y_t = \sum_{j=1}^{p} a_j y_{t-j} + \sum_{k=0}^{q} b_k e_{t-k} \tag{1.22}$$

where e_t is a white noise process. Using operator notation, we can write this in the form

$$A(q)y_t = (1 + B(q))\, e_t \tag{1.23}$$

where

$$A(q) = 1 - \sum_{j=1}^{p} a_j q^{-j} \tag{1.24}$$

$$B(q) = \sum_{k=1}^{q} b_k q^{-k} \tag{1.25}$$

and q^{-1} is the time delay operator $q^{-1}y_t = y_{t-1}$. $A(q)$ and $B(q)$ are thus operator polynomials.

We call ARMA modeling of time series y_t the procedure, in which the coefficients of the difference equation (1.22) are estimated based on the observations of y_t. If we have $B(q) \equiv 0$ the process (1.22) is called autoregressive, AR process. As a signal model the equation is called the AR(p) model.

Using the results of the linear filtering theory it can be shown that the power spectrum of ARMA(p, q) process is [54]

$$S(\omega) = \sigma_e^2 \frac{\left|1 + \sum_{k=1}^{q} b_k e^{-i\omega k}\right|^2}{\left|1 - \sum_{j=1}^{p} a_j e^{-i\omega j}\right|^2} \tag{1.26}$$

$$= \sigma_e^2 \frac{\left|1 + B(e^{-i\omega})\right|^2}{\left|A(e^{-i\omega})\right|^2} \tag{1.27}$$

CHAPTER II

Estimation theory

In this chapter we are conserned with estimation theory. The main goal of the chapter is to introduce the so-called mean square estimation scheme and give some results of its optimality in class of nonlinear and linear estimators. The aim of this introduction is to point out the importance of mean square criterion and motivate one of the topics of the next chapter: the recursive mean square estimation of time series parameters.

In this chapter we use z for vector of observations and θ for the parameters which are to be estimated. We use $\hat{\theta}$ for estimate of θ and $\hat{\theta}(z)$ for estimator, the function

$$\hat{\theta} = \hat{\theta}(z) \tag{2.1}$$

which connects the observations to the estimate. As estimation error, the difference between actual and estimated value of the parameter we use $\tilde{\theta} = \theta - \hat{\theta}$. Estimator for which the expectation of the estimation error is zero $E\left\{\tilde{\theta}\right\} = 0$ is said to be unbiased.

The estimator can be linear or nonlinear function of data. One of the main goals in this chapter is to find the conditions when the linear estimators are the best in class of all estimators. Linearity is also one of the main characteristic of the estimator when comparing the assumptions made for different estimators in this text.

Another main difference between the estimators is whether or not the parameters θ are treated as random. When θ is random we speak about Bayesian estimation and the goal is to find characteristics of the posterior density $p(\theta|z)$ of θ. Such estimates are e.g. mean square (MS) and maximum a posteriori (MAP) estimates. When θ is treated as unknown but non-random, we do not use any information about prior density of θ. Usually the estimation rule is then to minimize some estimation criterion. This criterion can still be probabilistic, as in Gauss–Markov estimate, because the estimator $\hat{\theta}(z)$ can still be random through the observation model. In some cases as in least squares estimation the estimation criterion is fully deterministic. We refer to non-random parameters with the term unknown parameter [61].

2.1 Observation model

We call the equation connecting the observations z to the parameters θ the observation model. For example

$$z = h(\theta, v) \tag{2.2}$$

where v is random measurement error, is a typical observation model. In most usual cases v is additive and the most general observation model used in this text is of the form

$$z = h(\theta) + v \tag{2.3}$$

In many cases we restrict our attention to linear observation models which can be written in form

$$z = H\theta + v \tag{2.4}$$

where H is matrix that does not contain parameters to be estimated. In all these cases θ can be random or fixed but unknown. In this text $\theta \in \mathbb{R}^p$ and $z, v \in \mathbb{R}^M$.

2.2 Maximum likelihood estimation

In maximum likelihood estimation the probability density function of the observation z given the unknown parameter θ is assumed to be known. No probability density of θ is required. For given data z, $\hat{\theta}_{\mathrm{ML}} = \hat{\theta}_{\mathrm{ML}}(z)$ is the maximum likelihood estimate, if

$$p(z|\hat{\theta}_{\mathrm{ML}}) \geq p(z|\hat{\theta}) \tag{2.5}$$

for any $\hat{\theta} \neq \hat{\theta}_{\mathrm{ML}}$. Thus, $\hat{\theta}_{\mathrm{ML}}$ maximizes the likelihood function $p(z|\theta)$ for given data z. In other words the selection of $\hat{\theta}_{\mathrm{ML}}$ makes the measurement z most probable in class of all probability densities which are of the form $p(z|\hat{\theta})$ [46, 33]. The maximum likelihood estimate is often thought to be the "true" estimate, when θ is treated as non-random. Other estimates are usually compared to the maximum likelihood estimate. The reason is that the maximum likelihood estimate has many desirable properties of a "good" estimate. For these see e.g. [9] and [33].

2.3 Bayes cost method

If we assume, that θ is a random vector having a known joint density $p(\theta, z)$ with the observation z we have made so-called Bayesian assumption [47]. This assumption leads to so-called Bayesian cost method for solving the estimator $\hat{\theta}(z)$. For cost method we define the function $C(\theta, \hat{\theta})$ which assigns to each combination of actual parameter value and estimate the unique cost. We call $C(\theta, \hat{\theta})$ the cost function. The expected value of the cost is given by

$$B(\hat{\theta}) = E\left\{C(\theta, \hat{\theta}(z))\right\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z))p(\theta, z)\, d\theta\, dz \tag{2.6}$$

From (1.9) we get $p(\theta, z) = p(z|\theta)p(\theta)$ and the expectation can be written in form

$$B(\hat{\theta}) = \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z))p(z|\theta)\, dz\right) p(\theta)\, d\theta \tag{2.7}$$

Obviously the inner integral is the conditioned expectation of the cost given θ and we write

$$\begin{aligned} B(\hat{\theta}|\theta) &= \int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z))p(z|\theta)\, dz && (2.8)\\ &= E\left\{C(\theta, \hat{\theta})\middle|\theta\right\} && (2.9) \end{aligned}$$

and in terms of the conditioned cost the Bayes cost of the estimator can be written as

$$\begin{aligned} B(\hat{\theta}) &= \int_{-\infty}^{\infty} B(\hat{\theta}|\theta)p(\theta)\, d\theta && (2.10)\\ &= E_\theta\left\{E\left\{C(\theta, \hat{\theta})\middle|\theta\right\}\right\} && (2.11)\\ &= E_\theta\left\{B(\hat{\theta}|\theta)\right\} && (2.12) \end{aligned}$$

Similarily using $p(\theta, z) = p(\theta|z)p(z)$ we can write

$$\begin{aligned} B(\hat{\theta}) &= E\left\{C(\theta, \hat{\theta}(z))\right\} && (2.13)\\ &= E_z\left\{E\left\{C(\theta, \hat{\theta})\middle|z\right\}\right\} && (2.14)\\ &= E_z\left\{B(\hat{\theta}|z)\right\} && (2.15) \end{aligned}$$

where

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z)) p(\theta|z)\, d\theta \tag{2.16}$$

is the conditional Bayes cost given z.

Now we can state the Bayes estimation criterion: For a given cost function $C(\theta, \hat{\theta})$, the Bayesian estimator $\hat{\theta}_{\mathrm{B}}$ is selected so that

$$B(\hat{\theta}_{\mathrm{B}}) \leq B(\hat{\theta}) \tag{2.17}$$

for any $\hat{\theta} \neq \hat{\theta}_{\mathrm{B}}$. So the Bayesian estimator is the one which minimizes the Bayesian cost. [46]

Different choices of the cost function lead to different estimators, and most common estimators can often be seen as minimizers of some specific cost function.

2.4 Mean-square -estimation

Next we study the situation, when the cost function is of the specific form, namely the squared norm of the estimation error $\tilde{\theta}$

$$C_{\mathrm{MS}}(\theta, \hat{\theta}) = \left\| \theta - \hat{\theta} \right\|^2 = \tilde{\theta}^T \tilde{\theta} \tag{2.18}$$

For that we rewrite (2.6) as

$$\begin{aligned} B(\hat{\theta}) &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z)) p(\theta|z)\, d\theta \right) p(z)\, dz & (2.19) \\ &= E_z \left\{ B(\hat{\theta}|z) \right\} & (2.20) \end{aligned}$$

where

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} C(\theta, \hat{\theta}(z)) p(\theta|z)\, d\theta \tag{2.21}$$

is called the conditional Bayes cost. Inserting the mean square cost function (2.18) to this we get the Bayesian mean square cost function

$$\begin{aligned} B_{\mathrm{MS}}(\hat{\theta}) &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \left\| \theta - \hat{\theta}(z) \right\|^2 p(\theta|z)\, d\theta \right) p(z)\, dz & (2.22) \\ &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \left(\theta^T \theta - 2\hat{\theta}^T(z)\theta + \hat{\theta}^T(z)\hat{\theta}(z) \right) p(\theta|z)\, d\theta \right) p(z)\, dz & (2.23) \end{aligned}$$

We define two functions as follows

$$\begin{aligned} \eta_{\theta|z}(z) &= \int_{-\infty}^{\infty} \theta p(\theta|z)\, d\theta & (2.24) \\ Q_{\theta|z}(z) &= \int_{-\infty}^{\infty} \left\| \theta - \eta_{\theta|z}(z) \right\|^2 p(\theta|z)\, d\theta & (2.25) \end{aligned}$$

which are the conditional mean and the conditional variance of θ given z, respectively, so that the terms in the inner integral of (2.23) can be rewritten

$$\begin{aligned} \int_{-\infty}^{\infty} \theta^T \theta p(\theta|z)\, d\theta &= Q_{\theta|z}(z) + \left\| \eta_{\theta|z}(z) \right\|^2 & (2.26) \\ \int_{-\infty}^{\infty} 2\hat{\theta}^T(z) \theta p(\theta|z)\, d\theta &= 2\hat{\theta}^T(z) \eta_{\theta|z}(z) & (2.27) \\ \int_{-\infty}^{\infty} \left\| \hat{\theta}^T(z) \right\|^2 p(\theta|z)\, d\theta &= \left\| \hat{\theta}^T(z) \right\|^2 & (2.28) \end{aligned}$$

Inserting to (2.23) we get Bayesian cost function

$$B_{\mathrm{MS}}(\hat{\theta}) = \int_{-\infty}^{\infty} Q_{\theta|z}(z)p(z)\,dz + \int_{-\infty}^{\infty} \left\|\eta_{\theta|z}(z) - \hat{\theta}(z)\right\|^2 p(z)\,dz \tag{2.29}$$

The first term in right hand side of the equation does not depend on $\hat{\theta}(z)$ and the second can be made to zero by choosing $\hat{\theta}(z) = \eta_{\theta|z}(z)$. Therefore we conclude that the optimal Bayesian minimum mean square estimator is the function $\eta_{\theta|z}(z)$, that is, the conditional mean

$$\hat{\theta}_{\mathrm{MS}} = \int_{-\infty}^{\infty} \theta p(\theta|z)\,d\theta = E\{\theta|z\} \tag{2.30}$$

This is, as seen, the conditional expectation of θ given the observation z. This result is independent of the density $p(\theta|z)$ [62]. The estimator $\hat{\theta}_{\mathrm{MS}}$ is sometimes also called the conditional mean estimator $\hat{\theta}_{\mathrm{CM}}$. The expected value of the estimation error $\tilde{\theta}$ can be written as

$$\begin{aligned}
E\left\{\tilde{\theta}\right\} &= \int_{-\infty}^{\infty} \tilde{\theta}p(\tilde{\theta})\,d\tilde{\theta} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{\theta}p(\tilde{\theta},z)\,d\tilde{\theta}\,dz && (2.31)\\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{\theta}p(\tilde{\theta}|z)p(z)\,d\tilde{\theta}\,dz && (2.32)\\
&= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} \tilde{\theta}p(\tilde{\theta}|z)\,d\tilde{\theta}\right]p(z)\,dz && (2.33)\\
&= E_z\left\{E\left\{\tilde{\theta}\middle|z\right\}\right\} && (2.34)\\
&= E_z\left\{E\left\{\theta - \hat{\theta}_{\mathrm{MS}}\middle|z\right\}\right\} && (2.35)\\
&= E_z\left\{\int_{-\infty}^{\infty}\theta p(\theta|z)\,dz - E\left\{\hat{\theta}_{\mathrm{MS}}\middle|z\right\}\right\} && (2.36)\\
&= E_z\left\{\hat{\theta}_{\mathrm{MS}} - E\left\{\hat{\theta}_{\mathrm{MS}}\middle|z\right\}\right\} && (2.37)
\end{aligned}$$

Now given z, $\hat{\theta}_{\mathrm{MS}}$ is deterministic and $E\left\{\hat{\theta}_{\mathrm{MS}}\middle|z\right\} = \hat{\theta}_{\mathrm{MS}}$, and

$$E\left\{\tilde{\theta}\right\} = 0 \tag{2.38}$$

This means that the mean square estimator is unbiased.

The preceeding results are easily modified to include a symmetric positive semidefinite (weighting) matrix W. We introduce so-called generalized mean square cost function

$$C_{\mathrm{GMS}}(\theta,\hat{\theta}) = \tilde{\theta}^T W\tilde{\theta} \tag{2.39}$$

For this the conditional Bayes cost is

$$\begin{aligned}
B(\hat{\theta}|z) &= E\left\{(\theta-\hat{\theta})^T W(\theta-\hat{\theta})\middle|z\right\} && (2.40)\\
&= \hat{\theta}^T W\hat{\theta} - 2\hat{\theta}^T W\eta_{\theta|z} + E\left\{\theta^T W\theta\middle|z\right\} && (2.41)\\
&= (\hat{\theta}-\eta_{\theta|z})^T W(\hat{\theta}-\eta_{\theta|z}) + E\left\{\theta^T W\theta\middle|z\right\} - \eta_{\theta|z}^T W\eta_{\theta|z} && (2.42)
\end{aligned}$$

Only the first term in right hand side of the equation depends on $\hat{\theta}$ and can be made equal to zero by choosing

$$\hat{\theta}_{\mathrm{GMS}} = E\{\theta|z\} \tag{2.43}$$

This result is identical to the result for $\hat{\theta}_{\mathrm{MS}}$.

The fact that the conditional mean minimizes the generalized index $C_{\mathrm{GMS}}(\theta,\hat{\theta})$ has an important implication. It means that the conditional mean minimizes the error in each component of $\tilde{\theta}$

individually [61]. This can be seen by choosing the weighting matrix W so that only one, say i'th, diagonal element is nonzero and equals to one and all the off diagonal elements are zero

$$W_i = \begin{pmatrix} 0 & \cdots & & & \cdots & 0 \\ \vdots & \ddots & & & & \vdots \\ & & 0 & & & \\ & & & 1 & & \\ & & & & 0 & \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & & & \cdots & 0 \end{pmatrix} \tag{2.44}$$

The minimization of the mean square cost function (2.18) can clearly be seen as minimization of the sum

$$C_{\mathrm{MS}}(\theta, \hat{\theta}) = \tilde{\theta}^T \tilde{\theta} = \sum_i \tilde{\theta}^T W_i \tilde{\theta} \tag{2.45}$$

and thus the conditional mean minimizes each squared error term $\tilde{\theta}_i$ individually. Note that the expectation of the estimation error is the variance of the estimate. Because this is minimized in mean square estimation, the mean square estimator is also called the minimum variance estimate $\hat{\theta}_{\mathrm{MV}}$.

2.5 Maximum a posteriori estimation

Let us now define another cost function, the uniform cost function

$$C_{\mathrm{UC}}(\theta, \hat{\theta}) = \begin{cases} 0 & , \tilde{\theta} \in I \\ 1 & , \text{otherwise} \end{cases} \tag{2.46}$$

where $I =]-\epsilon, \epsilon[\times \cdots \times]-\epsilon, \epsilon[\subset \mathbb{R}^p$ and ϵ is small. So this cost gives zero penalty if all components of the estimation error are small and a unit penalty if any of the components is larger than ϵ. When $C_{\mathrm{UC}}(\theta, \hat{\theta})$ is substituted into the equation of conditional Bayes cost (2.16) we obtain

$$B_{\mathrm{UC}}(\hat{\theta}|z) = \int_{\tilde{\theta} \in \overline{I}} p(\theta|z)\, d\theta \tag{2.47}$$

$$= 1 - \int_{\tilde{\theta} \in I} p(\theta|z)\, d\theta \tag{2.48}$$

where $\overline{I}$ states for the complement of I, and using the mean value theorem for integrals there is a value, say $\hat{\theta}$, in I for which

$$B_{\mathrm{UC}}(\hat{\theta}|z) = 1 - (2\epsilon)^p p(\hat{\theta}|z) \tag{2.49}$$

To minimize $B_{\mathrm{UC}}(\hat{\theta}|z)$ we must maximize $p(\hat{\theta}|z)$ so $\hat{\theta}_{\mathrm{UC}}$ can be defined by

$$p(\hat{\theta}_{\mathrm{UC}}|z) \geq p(\hat{\theta}|z) \tag{2.50}$$

forall $\hat{\theta} \neq \hat{\theta}_{\mathrm{UC}}$. Because $\hat{\theta}_{\mathrm{UC}}$ maximizes the posterior density of θ given observations z, $\hat{\theta}_{\mathrm{UC}}$ is also called the maximum a posteriori estimate $\hat{\theta}_{\mathrm{MAP}}$.

$\hat{\theta}_{\mathrm{UC}}$ is the statistical mode of density $p(\theta|z)$ and yet another name for the estimator is the conditional mode estimator. It can be shown, that if we assume that the prior distribution of θ is uniform in a region containing the maximum likelihood estimate, then the maximum likelihood estimate is identical to maximum a posteriori estimate, that is $\hat{\theta}_{\mathrm{ML}} = \hat{\theta}_{\mathrm{MAP}}$ [46]. Clearly $\hat{\theta}_{\mathrm{MAP}} = \hat{\theta}_{\mathrm{MS}}$ if the mode of the density $p(\theta|z)$ equals to the mean $\eta_{\theta|z}$. This is the case when $p(\theta|z)$ is symmetric and unimodal.

2.6 Linear minimum mean square estimator

In this section we restrict the form of the estimator to be a linear function of data and try to find the optimum estimator of that structure. If certain conditions for densities $p(\theta)$ and $p(z)$ are fulfilled, this optimal linear estimator turns to be overall optimum.

Suppose that the estimator is constrained to be a linear function of the data

$$\hat{\theta} = Kz \tag{2.51}$$

Let θ and z be random vectors with zero means and known covariances. No other assumptions are made about the joint distribution of parameters and data. We try to find the estimator that is of the form (2.51) and that minimizes the mean square Bayes cost $B_{\mathrm{MS}}(\hat{\theta})$. We first note that

$$\begin{aligned} B_{\mathrm{MS}}(\hat{\theta}) &= E\left\{\tilde{\theta}^T\tilde{\theta}\right\} && (2.52)\\ &= E\left\{(\theta-\hat{\theta})^T(\theta-\hat{\theta})\right\} && (2.53)\\ &= \operatorname{trace} E\left\{(\theta-\hat{\theta})(\theta-\hat{\theta})^T\right\} && (2.54)\\ &= \operatorname{trace} C_{\tilde{\theta}} && (2.55) \end{aligned}$$

and that

$$\begin{aligned} C_{\tilde{\theta}} &= E\left\{(\theta-\hat{\theta})(\theta-\hat{\theta})^T\right\} && (2.56)\\ &= E\left\{(\theta-Kz)(\theta-Kz)^T\right\} && (2.57)\\ &= C_\theta - KC_{z\theta} - C_{\theta z}K^T + KC_zK^T && (2.58)\\ &= C_\theta + (K-C_{\theta z}C_z^{-1})C_z(K-C_{\theta z}C_z^{-1})^T - C_{\theta z}C_z^{-1}C_{z\theta} && (2.59) \end{aligned}$$

Only the second term of the right hand side of the equation depends on the matrix K. The trace, and in fact each term of the diagonal (note that the diagonal is quadratic and thus positive), of the matrix $E\left\{(\theta-\hat{\theta})(\theta-\hat{\theta})^T\right\}$ can be minimized by choosing

$$K = C_{\theta z}C_z^{-1} \tag{2.60}$$

so that we get the linear minimum mean square estimate

$$\hat{\theta}_{\mathrm{LMS}} = C_{\theta z}C_z^{-1}z \tag{2.61}$$

and the estimation error covariance is from (2.59)

$$C_{\tilde{\theta}_{\mathrm{LMS}}} = C_\theta - C_{\theta z}C_z^{-1}C_{z\theta} \tag{2.62}$$

Using transformations

$$\begin{aligned} \theta' &= \theta - E\{\theta\} && (2.63)\\ z' &= z - E\{z\} && (2.64) \end{aligned}$$

result extends to variables with nonzero means [61]. This result means that the estimator (2.61) is optimum when its structure is restricted to be linear. The parameters are still thought to be random and no restrictions are made for the model of observations.

2.7 Minimum mean square estimator for Gaussian variables

We show next that if θ and z are jointly Gaussian, then the linear minimum variance estimator is not only the optimal linear estimator but the overall optimal estimator for θ. Suppose that the joint density function for θ and z is (without scaling terms)

$$p(\theta,z) \propto \exp\left\{-\frac{1}{2}\begin{pmatrix}\theta^T & z^T\end{pmatrix}\begin{pmatrix}C_\theta & C_{\theta z}\\ C_{z\theta} & C_z\end{pmatrix}^{-1}\begin{pmatrix}\theta\\ z\end{pmatrix}\right\} \tag{2.65}$$

For convenience θ and z have been assumed to have zero means. The nonzero mean situation can be treated with the transformations (2.63) and (2.64). It was shown that the mean square estimate equals to the conditional mean

$$\hat{\theta}_{\mathrm{MS}} = E\{\theta|z\} \tag{2.66}$$

For the calculation of the mean we first have to form the equation for the posterior density $p(\theta|z)$. First we note that the matrix inversion lemma [17] gives for the inverse of the joint covariance of θ and z

$$\begin{pmatrix} C_\theta & C_{\theta z} \\ C_{z\theta} & C_z \end{pmatrix}^{-1} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \tag{2.67}$$

where

$$\begin{aligned} C_{11} &= (C_\theta - C_{\theta z}C_z^{-1}C_{z\theta})^{-1} = C_\theta^{-1} + C_\theta^{-1}C_{\theta z}C_{22}C_{z\theta}C_\theta^{-1} && (2.68)\\ C_{22} &= (C_z - C_{z\theta}C_\theta^{-1}C_{\theta z})^{-1} = C_z^{-1} + C_z^{-1}C_{z\theta}C_{11}C_{\theta z}C_z^{-1} && (2.69)\\ C_{12} &= C_{21}^T = -C_{11}C_{\theta z}C_z^{-1} = -C_\theta^{-1}C_{\theta z}C_{22} && (2.70) \end{aligned}$$

The density of z can be written in the form

$$p(z) \propto \exp\left\{-\frac{1}{2}\begin{pmatrix} 0 & z^T \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & C_z^{-1} \end{pmatrix}\begin{pmatrix} 0 \\ z \end{pmatrix}\right\} \tag{2.71}$$

so that the posterior density $p(\theta|z)$ is obtained by forming

$$\begin{aligned} p(\theta|z) &= \frac{p(\theta, z)}{p(z)} && (2.72)\\ &\propto \exp\left\{-\frac{1}{2}\begin{pmatrix} \theta^T & z^T \end{pmatrix}\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} - C_z^{-1} \end{pmatrix}\begin{pmatrix} \theta \\ z \end{pmatrix}\right\} && (2.73)\\ &= \exp\left\{-\frac{1}{2}\left(\theta^T C_{11}\theta + 2\theta^T C_{12}z + z^T(C_{22} - C_z^{-1})z\right)\right\} && (2.74)\\ &= \exp\left\{-\frac{1}{2}\left(\theta^T C_{11}\theta + 2\theta^T C_{11}C_{\theta z}C_z^{-1}z + z^T C_z^{-1}C_{z\theta}C_{11}C_{\theta z}C_z^{-1}z\right)\right\} && (2.75)\\ &= \exp\left\{-\frac{1}{2}(\theta^T - C_{\theta z}C_z^{-1}z)^T C_{11}(\theta^T - C_{\theta z}C_z^{-1}z)\right\} && (2.76) \end{aligned}$$

This is clearly a Gaussian density. The Gaussian conditional density is of the form

$$p(\theta|z) \propto \exp\left\{-\frac{1}{2}(\theta - E\{\theta|z\})^T C_{\theta|z}^{-1}(\theta - E\{\theta|z\})\right\} \tag{2.77}$$

Because $\hat{\theta}_{\mathrm{MS}} = E\{\theta|z\}$, $C_{\theta|z} = C_{\tilde{\theta}_{\mathrm{MS}}}$ and comparing with (2.76) we can conclude

$$\begin{aligned} \hat{\theta}_{\mathrm{MS}} &= C_{\theta z}C_z^{-1}z && (2.78)\\ C_{\tilde{\theta}_{\mathrm{MS}}} &= C_\theta - C_{\theta z}C_z^{-1}C_{z\theta} && (2.79) \end{aligned}$$

which is exactly the linear minimum mean square estimator.

2.8 Mean square estimation with observation model

This far the form of the mean square estimator has been independent of the observation model. The calculation of cross covariances is however not possible without a model for the dependence between observations and parameters. A popular model is the so-called additive noise model

$$z = h(\theta) + v \tag{2.80}$$

Now we have to assume some joint density $p(\theta, v)$ for random parameters θ and observation error v. Then, at least theoretically, the joint density of z and θ is known and we can form either $p(z|\theta)$ if we are going to calculate the maximum likelihood estimate or using Bayes rule the posterior density $p(\theta|z)$ of θ if a Bayesian estimate is to be calculated. In general the density $p(\theta|z)$ is needed e.g. for the mean square estimator, but in certain situtions, as in the case of linear estimates the knowledge about the second order statistics is enough. A detailed example of Bayesian and maximum likelihood estimation with general observation model is calculated in section 2.11.

Let us now constrain the observations to be of specific linear form of parameters

$$z = H\theta + v \tag{2.81}$$

where v and θ are random. Suppose that θ and v have zero means and known covariances. z is then zero mean and has covariance

$$\begin{aligned} C_z &= E\left\{(H\theta + v)(H\theta + v)^T\right\} && (2.82)\\ &= HC_\theta H^T + HC_{\theta v} + C_{v\theta}H^T + C_v && (2.83)\end{aligned}$$

and the cross covariance $C_{\theta z}$ is

$$C_{\theta z} = E\left\{\theta(H\theta + v)^T\right\} = C_\theta H^T + C_{\theta v} \tag{2.84}$$

Using these for linear mean square estimate we get

$$\hat{\theta}_{\text{LMS}} = (C_\theta H^T + C_{\theta v})(HC_\theta H^T + HC_{\theta v} + C_{v\theta}H^T + C_v)^{-1} z \tag{2.85}$$

with error covariance matrix

$$C_{\tilde{\theta}_{\text{LMS}}} = C_\theta - (C_\theta H^T + C_{\theta v})(HC_\theta H^T + HC_{\theta v} + C_{v\theta}H^T + C_v)^{-1}(HC_\theta + C_{v\theta}) \tag{2.86}$$

A special case of this is when θ and v are uncorrelated, $C_{\theta v} = C_{v\theta} = 0$. Then the equations for the estimate and error reduce to

$$\begin{aligned} \hat{\theta}_{\text{LMS}} &= C_\theta H^T (HC_\theta H^T + C_v)^{-1} z && (2.87)\\ C_{\tilde{\theta}_{\text{LMS}}} &= C_\theta - C_\theta H^T (HC_\theta H^T + C_v)^{-1} HC_\theta && (2.88)\end{aligned}$$

Applying the augmented matrix inversion lemma we get

$$\begin{aligned} \hat{\theta}_{\text{LMS}} &= (C_\theta^{-1} + H^T C_v^{-1} H)^{-1} H^T C_v^{-1} z = C_{\tilde{\theta}_{\text{LMS}}} H^T C_v^{-1} z && (2.89)\\ C_{\tilde{\theta}_{\text{LMS}}} &= (C_\theta^{-1} + H^T C_v^{-1} H)^{-1} && (2.90)\end{aligned}$$

2.9 Gauss-Markov estimate

Next we consider the problem of estimating the unknown (non-random) parameters θ as linear function of data. Suppose, that

$$z = H\theta + v \tag{2.91}$$

where v is random. θ is non-random but unknown. Let v have zero mean and covariance C_v. We try to find the linear unbiased estimator $\hat{\theta}$ that minimizes the mean square criterion. Let

$$\hat{\theta} = Kz + k \tag{2.92}$$

with the requirement

$$E\left\{\hat{\theta}\right\} = \theta \tag{2.93}$$

Then

$$\begin{aligned} E\left\{\hat{\theta}\right\} &= E\{Kz + k\} = KE\{z\} + k && (2.94)\\ &= KE\{H\theta + v\} + k = KH\theta + k && (2.95)\end{aligned}$$

For $\hat{\theta}$ to be unbiased K and k must satisfy

$$KH = I, \quad k = 0 \tag{2.96}$$

For the error covariance we get

$$\begin{aligned} C_{\tilde{\theta}} &= E\left\{(\theta - \hat{\theta})(\theta - \hat{\theta})^T\right\} && (2.97)\\ &= E\left\{(\theta - Kz)(\theta - Kz)^T\right\} && (2.98)\\ &= E\left\{(\theta - KH\theta - Kv)(\theta - KH\theta - Kv)^T\right\} && (2.99)\\ &= E\left\{Kvv^TK^T\right\} = KC_vK^T && (2.100) \end{aligned}$$

Next we consider the matrix

$$K' = (H^TC_v^{-1}H)^{-1}H^TC_v^{-1} \tag{2.101}$$

for which $K'H = I$. With this selection the estimator $\hat{\theta} = K'z$ is unbiased. We will next show that K' minimizes the mean square error. First we note that

$$\begin{aligned} K'C_vK^T &= (H^TC_v^{-1}H)^{-1}H^TC_v^{-1}C_vK^T && (2.102)\\ &= (H^TC_v^{-1}H)^{-1}(KH)^T && (2.103)\\ &= (H^TC_v^{-1}H)^{-1} && (2.104) \end{aligned}$$

$$\begin{aligned} KC_vK'^T &= KC_vC_v^{-1}H(H^TC_v^{-1}H)^{-1} && (2.105)\\ &= (KH)(H^TC_v^{-1}H)^{-1} && (2.106)\\ &= (H^TC_v^{-1}H)^{-1} && (2.107) \end{aligned}$$

and

$$\begin{aligned} K'C_vK'^T &= (H^TC_v^{-1}H)^{-1}H^TC_v^{-1}C_vC_v^{-1}H(H^TC_v^{-1}H^T)^{-1} && (2.108)\\ &= (H^TC_v^{-1}H)^{-1}H^TC_v^{-1}H(H^TC_v^{-1}H^T)^{-1} && (2.109)\\ &= (H^TC_v^{-1}H)^{-1} && (2.110) \end{aligned}$$

Next we form the matrix

$$\begin{aligned} (K - K')C_v(K - K')^T &= KC_vK^T - K'C_vK^T - KC_vK'^T + K'C_vK'^T && (2.111)\\ &= KC_vK^T - K'C_vK'^T && (2.112) \end{aligned}$$

Note that the matrix $(K - K')C_v(K - K')^T$ is positive semidefinite. Now we get for the trace of the error covariance matrix

$$\begin{aligned} \operatorname{trace} C_{\tilde{\theta}} &= \operatorname{trace} KC_vK^T && (2.113)\\ &= \operatorname{trace}(K - K')C_v(K - K')^T + \operatorname{trace} K'C_vK'^T && (2.114) \end{aligned}$$

The trace, and in fact every diagonal element separately, can be minimized by choosing $K = K'$. Thus we get for Gauss–Markov estimate

$$\begin{aligned} \hat{\theta}_{\mathrm{GM}} &= (H^TC_v^{-1}H)^{-1}H^TC_v^{-1}z && (2.115)\\ C_{\tilde{\theta}_{\mathrm{GM}}} &= (H^TC_v^{-1}H)^{-1} && (2.116) \end{aligned}$$

Comparing $\hat{\theta}_{\mathrm{GM}}$ with $\hat{\theta}_{\mathrm{LMS}}$ we note that $\hat{\theta}_{\mathrm{GM}}$ is obtained by letting $C_\theta^{-1} = 0$. This means that the prior density of θ is flat or no *a priori* information is available about θ [13, 6].

Note that the criterion to be minimized in Gauss–Markov estimation is identical to that of mean square estimator. The difference is in treatment of θ as non-random but unknown parameter vector.

Because this estimator also minimizes the variance of the estimator it is also called the linear minimum variance estimator $\hat{\theta}_{\mathrm{LMV}}$.

2.10 Least squares estimation

Finally we treat the situation, where neither the parameters θ or the error v in observations is interpreted as random. The solution of this problem leads to (generalized or weighted) least squares solution.

Let the observation model be

$$z = h(\theta) + v \tag{2.117}$$

where θ and v are unknown but non-random. Then the generalized (weighted) least squares estimator $\hat{\theta}_{\mathrm{GLS}}$ is defined to be the minimizer of the generalized least squares index

$$\begin{aligned} l_{\mathrm{GLS}} &= (z - h(\theta))W(z - h(\theta))^T && (2.118)\\ &= \|Lz - Lh(\theta)\|^2 && (2.119) \end{aligned}$$

where $L^T L = W$. W is a symmetric positive definite matrix.

With a nonlinear $h(\theta)$, the minimization can be done e.g. with Gauss–Newton algorithm. We first form the Taylor expansion of the generalized least squares index in neighborhood of θ^*

$$\begin{aligned} l(\theta) &= l(\theta^*) + \left(\frac{\partial l}{\partial \theta}(\theta^*)\right)(\theta - \theta^*) && (2.120)\\ &+ \frac{1}{2}(\theta - \theta^*)^T \left(\frac{\partial^2 l}{\partial \theta^2}(\theta^*)\right)(\theta - \theta^*) && (2.121)\\ &+ O(\|\theta - \theta^*\|^2) && (2.122) \end{aligned}$$

Note that $\partial l/\partial\theta$ is a row vector and and $\partial^2 l/\partial\theta^2$ is a symmetric matrix. Next we approximate $l(\theta)$ with the second order approximation

$$l(\theta) \approx l(\theta^*) + \left[\left(\frac{\partial l}{\partial \theta}(\theta^*)\right) + \frac{1}{2}(\theta - \theta^*)^T \left(\frac{\partial^2 l}{\partial \theta^2}(\theta^*)\right)\right](\theta - \theta^*) = f(\theta) \tag{2.123}$$

This approximation is at minimum, when $\theta = \hat{\theta}$ so, that

$$\frac{\partial f}{\partial \theta}(\hat{\theta}) = \left(\frac{\partial l}{\partial \theta}(\theta^*)\right) + \frac{1}{2}(\hat{\theta} - \theta^*)^T \left(\frac{\partial^2 l}{\partial \theta^2}(\theta^*)\right) = 0 \tag{2.124}$$

so that we can solve $\hat{\theta}$

$$\hat{\theta} = \theta^* - 2\left(\frac{\partial^2 l}{\partial \theta^2}(\theta^*)\right)^{-1} \left(\frac{\partial l}{\partial \theta}(\theta^*)\right)^T \tag{2.125}$$

The gradient of l can be formed

$$\frac{\partial l}{\partial \theta}(\theta^*) = -2(z - h(\theta^*))^T W \frac{\partial h}{\partial \theta}(\theta^*) \tag{2.126}$$

and differentiating twice, we get

$$\frac{\partial^2 l}{\partial \theta^2}(\theta^*) = -2\left(\sum_{i=1}^{M}(z_i - h_i(\theta^*))W\frac{\partial^2 h_i}{\partial \theta^2}(\theta^*)\right) + 2\left(\frac{\partial h}{\partial \theta}(\theta^*)\right)^T W \left(\frac{\partial h}{\partial \theta}(\theta^*)\right) \tag{2.127}$$

Finally, we obtain a recursion

$$\hat{\theta}_{i+1} = \hat{\theta}_i + k\left(J_i^T W J_i - \sum_{j=1}^{M}(z_j - h_j(\hat{\theta}_i))W\frac{\partial^2 h_j}{\partial \theta^2}(\hat{\theta}_i)\right)^{-1} \left(J_i^T W(z - h(\hat{\theta}_i))\right) \tag{2.128}$$

where $J_i = \frac{\partial h}{\partial \theta}(\hat{\theta}_i)$. k is the step size parameter, which ensures that the recursion converges at least to local minimum [3]. This method is called the Newton–Raphson method. If the norm $\|z - h(\theta)\|$ is small we can make the approximation

$$\frac{\partial^2 l}{\partial \theta^2}(\hat{\theta}_i) = 2J_i^T W J_i \tag{2.129}$$

and the iteration gets the form

$$\hat{\theta}_{i+1} = \hat{\theta}_i + k \left(J_i^T W J_i\right)^{-1} \left(J_i^T W (z - h(\hat{\theta}_i))\right) \tag{2.130}$$

This is called the Gauss-Newton method. Yet another well known search procedure results, when $(\partial^2 l/\partial\theta^2)(\hat{\theta}_i)$ is replaced with the identity matrix. With $W = I$ we can write

$$\hat{\theta}_{i+1} = \hat{\theta}_i + k \left(J_i^T (z - h(\hat{\theta}_i))\right) \tag{2.131}$$

This is called the steepest descent method.

If $h(\theta)$ is linear the observation model is of the form

$$z = H\theta + v \tag{2.132}$$

Now the minimization of the generalized linear least squares index

$$\begin{aligned} l_{\mathrm{GLS}} &= (z - H\theta)W(z - H\theta)^T & (2.133)\\ &= \|Lz - LH\theta\|^2 & (2.134) \end{aligned}$$

takes a simpler form. Let first $W = I$ and denote the corresponding index with l_{LS}. Then form the gradient

$$\frac{\partial l_{\mathrm{LS}}}{\partial \theta} = -2(z - H\theta)^T H \tag{2.135}$$

Then, the least squares estimator satisfies

$$-(z - H\hat{\theta}_{\mathrm{LS}})^T H = 0 \tag{2.136}$$

or

$$H^T(z - H\hat{\theta}_{\mathrm{LS}}) = 0 \tag{2.137}$$

This can be rewritten in the form

$$H^T H\hat{\theta}_{\mathrm{LS}} = H^T z \tag{2.138}$$

This system of equations is called the normal equations. The formal solution is

$$\hat{\theta}_{\mathrm{LS}} = (H^T H)^{-1} H^T z \tag{2.139}$$

Note that the matrix $H^T H$ equals to $\partial^2 l_{\mathrm{LS}}/\partial\theta^2$ so that if it is positive definite, the solution is guaranteed to be the stationary point of l_{LS}.

The generalized solution is obtained by multiplying the equation with matrix L

$$Lz = LH\theta + Lv \tag{2.140}$$

and with notations $z' = Lz$ and $H' = LH$

$$z' = H'\theta + Lv \tag{2.141}$$

Now the minimization of the generalized index

$$l_{\mathrm{GLS}} = (z' - H'\theta)W(z' - H'\theta)^T \tag{2.142}$$

is obtained by using (2.139)

$$\begin{aligned} \hat{\theta}_{\mathrm{GLS}} &= (H'^T H')^{-1} H'^T z' & (2.143)\\ &= (H^T L^T LH)^{-1} H^T L^T Lz & (2.144)\\ &= (H^T W H)^{-1} H^T W z & (2.145) \end{aligned}$$

This is seen to be equivalent to the Gauss–Markov estimate $\hat{\theta}_{\mathrm{GM}}$ if we choose $W = C_v^{-1}$.

A classical reference for linear and nonlinear least squares problems is [37] and for nonlinear optimization generally [27].

2.11 Solution of ML, MAP and MS estimates

As a final example we compare the maximum likelihood estimation with maximum a posteriori estimation. Assume that we have following model for obseravation

$$z = h(\theta) + v \tag{2.146}$$

where θ and v are random parameters. Only the parameters θ are to be estimated. Given θ, the observations z and the error v have the same density, except that the mean $E\{z\} = E\{v\} + h(\theta)$. The density of z given θ is thus

$$p(z|\theta) = p_v(z - h(\theta)|\theta) \tag{2.147}$$

Assuming v is Gaussian with zero mean and θ and v are independent we get

$$p(z|\theta) \propto \exp\left\{-\frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta))\right\} \tag{2.148}$$

and taking logarithms

$$\log p(z|\theta) = \text{const} - \frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta)) \tag{2.149}$$

Maximization of this so-called log-likelihood function gives the maximum likelihood estimate. Note that maximixation of $\log p(z|\theta)$ is identical to to minimization of

$$l_{\text{ML}} = \frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta)) \tag{2.150}$$

This is identical to generalized least squares estimation. Note that in the general case the minimization is nonlinear and can be done e.g. with Gauss-Newton algorithm.

The posterior density is now

$$\begin{aligned} p(\theta|z) &\propto p(z|\theta)p(\theta) && (2.151)\\ &= p_v(z - h(\theta)|\theta)p(\theta) && (2.152) \end{aligned}$$

Let now $v \sim \mathcal{N}(0, C_v)$ and $\theta \sim \mathcal{N}(\eta_\theta, C_\theta)$, then

$$\begin{aligned} p(\theta|z) &\propto \exp\left\{-\frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta))\right\}p(\theta) && (2.153)\\ &\propto \exp\left\{-\frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta))\right\} \exp\left\{-\frac{1}{2}(\theta - \eta_\theta)^T C_\theta^{-1}(\theta - \eta_\theta)\right\} && (2.154) \end{aligned}$$

The maximum a posteriori estimate is obtained by maximizing the log likelihood function

$$\log p(\theta|z) = \text{const} - \frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta)) - \frac{1}{2}(\theta - \eta_\theta)^T C_\theta^{-1}(\theta - \eta_\theta) \tag{2.155}$$

This is seen to be formally identical to generalized least squares estimation with

$$l_{\text{MAP}} = \frac{1}{2}(z - h(\theta))^T C_v^{-1}(z - h(\theta)) + \frac{1}{2}(\theta - \eta_\theta)^T C_\theta^{-1}(\theta - \eta_\theta) \tag{2.156}$$

The quadratic index (2.156) can be written in form

$$\begin{aligned} 2l_{\text{MAP}} &= \left((z - h(\theta))^T, (\theta - \eta_\theta)^T\right) \begin{pmatrix} C_v^{-1} & 0 \\ 0 & C_\theta^{-1} \end{pmatrix} \begin{pmatrix} z - h(\theta) \\ \theta - \eta_\theta \end{pmatrix} && (2.157)\\ &= \left\| L \begin{pmatrix} z \\ \eta_\theta \end{pmatrix} - L \begin{pmatrix} h(\theta) \\ \theta \end{pmatrix} \right\|^2 && (2.158) \end{aligned}$$

where

$$L^T L = W, \quad W = \begin{pmatrix} C_v^{-1} & 0 \\ 0 & C_\theta^{-1} \end{pmatrix} \tag{2.159}$$

If we assume the linear observation model $h(\theta) = H\theta$ we get

$$2l_{\mathrm{MAP}} = \left\| L \begin{pmatrix} z \\ \eta_\theta \end{pmatrix} - L \begin{pmatrix} H\theta \\ \theta \end{pmatrix} \right\|^2 \tag{2.160}$$

$$= \|Lz' - LH'\theta\|^2 \tag{2.161}$$

where

$$H' = \begin{pmatrix} H \\ I \end{pmatrix}, \quad z' = \begin{pmatrix} z \\ \eta_\theta \end{pmatrix} \tag{2.162}$$

This can be solved as generalized linear LS problem and the formal solution is

$$\hat{\theta}_{\mathrm{GLS}} = (H'^T L^T L H')^{-1} H'^T L^T L z' \tag{2.163}$$

$$= (H'^T W H')^{-1} H'^T W z' \tag{2.164}$$

$$= (H^T C_v^{-1} H + C_\theta^{-1})^{-1} (H^T C_v^{-1} z + C_\theta^{-1} \eta_\theta) \tag{2.165}$$

This is seen to be equivalent with the linear mean square estimate. Note that this form contains the nonzero mean η_θ of θ. If we further assume that $C_\theta^{-1} = 0$ corresponding to infinite variance, we get exactly the Gauss–Markov estimate. As seen this is equivalent to the maximum likelihood estimate. If we have no information about the observation error v, we can assume $C_v^{-1} = I$ and the form of the estimator is

$$\hat{\theta} = (H^T H)^{-1} H^T z \tag{2.166}$$

which is identical to the linear least squares solution.

2.12 Optimality of the MS estimator

The selection of the cost function is a somewhat arbitrary process. In this section we show that the mean square criterion is optimal in many sense.

First we recall that the Bayes cost function can be written in form (2.20)

$$B(\hat{\theta}) = E_z \left\{ E \left\{ C(\theta, \hat{\theta}) \middle| z \right\} \right\} = E_z \left\{ B(\hat{\theta}|z) \right\} \tag{2.167}$$

where the outer expectation does not depend on θ so that $B(\hat{\theta})$ is minimized by minimizing the conditional bayes cost $B(\hat{\theta}|z) = E \left\{ C(\theta, \hat{\theta}) \middle| z \right\}$.

Let now $L(\tilde{\theta})$ be a cost function which is function of estimation error $\tilde{\theta}$ only. That is $C(\theta, \hat{\theta}) = L(\theta - \hat{\theta})$. Let L be symmetric about $\tilde{\theta} = 0$

$$L(\tilde{\theta}) = L(-\tilde{\theta}) \tag{2.168}$$

and convex

$$L(\lambda\tilde{\theta}_1 + (1-\lambda)\tilde{\theta}_2) \leq \lambda L(\tilde{\theta}_1) + (1-\lambda) L(\tilde{\theta}_2), \quad 0 \leq \lambda \leq 1 \tag{2.169}$$

These properties cover wide range of cost functions and estimators. For example the quadratic cost function

$$L_1(\tilde{\theta}) = \tilde{\theta}^T \tilde{\theta} \tag{2.170}$$

and the absolute error cost function

$$L_2(\tilde{\theta}) = \sum \left| \tilde{\theta}_i \right| \tag{2.171}$$

belong to this class of cost functions.

Let $\hat{\theta}_{\mathrm{MS}}$ denote the unbiased estimator that minimizes the mean square error. Then if the posterior density $p(\theta|z)$ is symmetric about $\hat{\theta}_{\mathrm{MS}}$, the estimator minimizing L is identical to $\hat{\theta}_{\mathrm{MS}}$. This can be seen as follows. For any estimator $\hat{\theta}$ the conditional Bayes cost is

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} L(\theta - \hat{\theta}) p(\theta|z)\, d\theta \tag{2.172}$$

Now let $\xi = \theta - \hat{\theta}_{\mathrm{MS}}$, then

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} L(\xi + \hat{\theta}_{\mathrm{MS}} - \hat{\theta}) p(\hat{\theta}_{\mathrm{MS}} + \xi|z)\, d\xi \tag{2.173}$$

This must hold for $-\xi$ too

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} L(-\xi + \hat{\theta}_{\mathrm{MS}} - \hat{\theta}) p(\hat{\theta}_{\mathrm{MS}} - \xi|z)\, d\xi \tag{2.174}$$

Now, L is symmetric about 0 and p is symmetric about $\hat{\theta}_{\mathrm{MS}}$, thus we get

$$B(\hat{\theta}|z) = \int_{-\infty}^{\infty} L(\xi - \hat{\theta}_{\mathrm{MS}} + \hat{\theta}) p(\hat{\theta}_{\mathrm{MS}} + \xi|z)\, d\xi \tag{2.175}$$

Adding (2.173) and (2.175) and using the convexity of L we get

$$\begin{aligned} B(\hat{\theta}|z) &= \int_{-\infty}^{\infty} \left[\frac{1}{2} L(\xi + \hat{\theta}_{\mathrm{MS}} - \hat{\theta}) + \frac{1}{2} L(\xi - \hat{\theta}_{\mathrm{MS}} + \hat{\theta})\right] p(\hat{\theta}_{\mathrm{MS}} + \xi|z)\, d\xi && (2.176)\\ &\geq \int_{-\infty}^{\infty} L\left(\frac{1}{2}(\xi + \hat{\theta}_{\mathrm{MS}} - \hat{\theta}) + \frac{1}{2}(\xi - \hat{\theta}_{\mathrm{MS}} + \hat{\theta})\right) p(\hat{\theta}_{\mathrm{MS}} + \xi|z)\, d\xi && (2.177)\\ &= \int_{-\infty}^{\infty} L(\xi) p(\hat{\theta}_{\mathrm{MS}} + \xi|z)\, d\xi && (2.178) \end{aligned}$$

and substituting $\xi = \theta - \hat{\theta}_{\mathrm{MS}}$ we get

$$\begin{aligned} B(\hat{\theta}|z) &\geq \int_{-\infty}^{\infty} L(\theta - \hat{\theta}_{\mathrm{MS}}) p(\theta|z)\, d\theta && (2.179)\\ &= B(\hat{\theta}_{\mathrm{MS}}|z) && (2.180) \end{aligned}$$

This means that the conditional Bayes cost associated with any estimator can never be less than that associated with the MS estimator. Any estimator that minimizes this kind of a cost function must be equal to $\hat{\theta}_{\mathrm{MS}}$. For conditions (2.168) and (2.169) MS estimator is thus optimum. [61, 46]

2.13 Recursive mean square estimation

In the preceding chapters we have dealt with systems, where some fixed amount of data has been available for estimation of the unknown or random parameters. The estimators have been functions of the whole data set. In recursive estimation the question arises how to update the estimate, when some amount of new data is received. The answer for this question with certain observation models is the so-called Kalman filtering approach. Kalman filter is a tool with which one can estimate the sequence of the states of the dynamical system that cannot be observed directly. The available information is the data sequence that carries some information about the states. algorithm.

In this context we introduce the state-space model for linear dynamical systems. Let $y_t \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^p$ be vector-valued processes. The state x_t evolves according to linear difference equation

$$x_{t+1} = F_t x_t + G_t w_t \tag{2.181}$$

with some initial distribution for x_0. The state cannot be observed directly. Instead, measurements y_t are available at discrete sampling times and are described as

$$y_t = H_t x_t + v_t \tag{2.182}$$

v_t and w_t are white noise sequences. This is clearly a linear observation model. The other assumptions for the model are as follows

- F_t, G_t and H_t are known sequences of matrices.
- (x_0, w_t, v_t) is a sequence of mutually uncorrelated random vectors with finite variance.
- $E\{w_t\} = 0,\ E\{v_t\} = 0,\ \forall t$
- covariances C_{w_t}, C_{v_t} and C_{v_t,w_t} are known sequences of matrices.

The Kalman filtering problem is now to find the linear minimum mean square estimator $\hat{x}_t$ for state x_t given the observations $y_1, \ldots, y_t$. This has been shown to be equal to conditional mean

$$\hat{x}_t = E\{x_t | y_1, \ldots y_t\} \tag{2.183}$$

In derivation of Kalman filter we also require that the estimate is recursive (sequential). The new estimate is to be based on the new data and the preceding estimate implicitly. Note that we use here notations y_t and x_t for z_t and θ_t respectively. This is done for historical reasons.

As we have seen we can use two approaches to obtain the linear mean square estimate. The first is to specify a linear conditional mean and find the best linear form. The second approach is to assume, that x_t, v_t and w_t are Gaussian. In this case the conditional mean is again linear. The results of these two approaches are identical. In other words Kalman filter is the best sequential estimator if the Gaussian assumption is valid and it is the best linear estimator whatever the distributions are.

We employ the second approach. We use the fact that the maximum a posteriori and mean square estimates are identical as seen in section 2.11. First we have to look at the expression for the density of x_t given $y_t, \ldots, y_1$. We make the notation $Y_t = (y_t, \ldots, y_1)$. Then

$$p(x_t | y_t, \ldots, y_1) = \frac{p(x_t, Y_t)}{p(Y_t)} \tag{2.184}$$

$$= \frac{p(x_t, y_t, Y_{t-1})}{p(y_t, Y_{t-1})} \tag{2.185}$$

For the numerator we can write

$$p(x_t, y_t, Y_{t-1}) = p(y_t | x_t, Y_{t-1}) p(x_t, Y_{t-1}) \tag{2.186}$$

$$= p(y_t | x_t, Y_{t-1}) p(x_t | Y_{t-1}) p(Y_{t-1}) \tag{2.187}$$

If x_t is given, the only random term in (2.182), is v_t that does not depend on x_t or Y_{t-1}. Thus

$$p(y_t | x_t, Y_{t-1}) = p(y_t | x_t) \tag{2.188}$$

and we can write

$$p(x_t | Y_t) = \frac{p(y_t | x_t) p(x_t | Y_{t-1}) p(Y_{t-1})}{p(y_t, Y_{t-1})} \tag{2.189}$$

$$= \frac{p(y_t | x_t) p(x_t | Y_{t-1}) p(Y_{t-1})}{p(y_t | Y_{t-1}) p(Y_{t-1})} \tag{2.190}$$

$$= \frac{p(y_t | x_t) p(x_t | Y_{t-1})}{p(y_t | Y_{t-1})} \tag{2.191}$$

Now because each of the densities in right hand side is Gaussian, we need only to form the means and the covariances of the densities to find the density of x_t given Y_t.

For $p(y_t|x_t)$ we get

$$\begin{aligned} E\{y_t|x_t\} &= E\{H_t x_t + v_t|x_t\} = H_t x_t & (2.192)\\ C_{y_t|x_t} &= E\left\{(H_t x_t + v_t - E\{y_t|x_t\})(H_t x_t + v_t - E\{y_t|x_t\})^T|x_t\right\} = C_{v_t} & (2.193)\end{aligned}$$

For $p(x_t|Y_{t-1})$ we get

$$\begin{aligned} E\{x_t|Y_{t-1}\} &= E\{F_{t-1}x_{t-1} + G_{t-1}w_{t-1}|Y_{t-1}\} & (2.194)\\ &= F_{t-1}\hat{x}_{t-1} = \hat{x}_{t|t-1} & (2.195)\end{aligned}$$

$\hat{x}_{t|t-1}$ is thus the prediction of x_t based on $\hat{x}_{t-1}$, $\hat{x}_{t-1} = E\{x_{t-1}|Y_{t-1}\}$ is the optimal MS estimate at time $t-1$ and

$$\begin{aligned} C_{x_t|Y_{t-1}} &= E\left\{(x_t - \hat{x}_{t|t-1})(x_t - \hat{x}_{t|t-1})^T|Y_{t-1}\right\} & (2.196)\\ &= E\left\{\tilde{x}_{t|t-1}\tilde{x}_{t|t-1}^T|Y_{t-1}\right\} = C_{\tilde{x}_{t|t-1}} & (2.197)\end{aligned}$$

Next we could form the mean and covariance of $p(y_t|Y_{t-1})$, form the density $p(x_t|Y_t)$ and calculate the conditional mean $E\{x_t|Y_t\}$ which is the estimator we are looking for. However to obtain the MAP estimator we only need to maximize the numerator of (2.191) or the logarithm of that. Now clearly

$$\log p(x_t|Y_t) \propto \text{const} - (y_t - H_t x_t)C_{v_t}^{-1}(y_t - H_t x_t)^T - (x_t - \hat{x}_{t|t-1})C_{\tilde{x}_{t|t-1}}^{-1}(x_t - \hat{x}_{t|t-1})^T \quad (2.198)$$

The derivative of this with respect to x_t equals to zero when $x_t = \hat{x}_t$

$$H_t^T C_{v_t}^{-1}(y_t - H_t\hat{x}_t) - C_{\tilde{x}_{t|t-1}}^{-1}(\hat{x}_t - \hat{x}_{t|t-1}) = 0 \quad (2.199)$$

Solving $\hat{x}_t$ gives the MAP estimate

$$\hat{x}_t = (H_t^T C_{v_t}^{-1} H_t + C_{\tilde{x}_{t|t-1}}^{-1})^{-1}(C_{\tilde{x}_{t|t-1}}^{-1}\hat{x}_{t|t-1} + H_t^T C_{v_t}^{-1} y_t) \quad (2.200)$$

This is clearly seen to be of the form (2.165), the Bayesian MAP estimate using the last available estimate as mean of x_t. The first term on right hand side of the equation can be expanded using the matrix inversion lemma

$$\begin{aligned} \hat{x}_t &= (C_{\tilde{x}_{t|t-1}} - C_{\tilde{x}_{t|t-1}}H_t^T(H_t C_{\tilde{x}_{t|t-1}}H_t^T + C_{v_t})^{-1}H_t C_{\tilde{x}_{t|t-1}}) & (2.201)\\ &\quad (C_{\tilde{x}_{t|t-1}}^{-1}\hat{x}_{t|t-1} + H_t^T C_{v_t}^{-1} y_t) & (2.202)\\ &= (C_{\tilde{x}_{t|t-1}} - K_t H_t C_{\tilde{x}_{t|t-1}})(C_{\tilde{x}_{t|t-1}}^{-1}\hat{x}_{t|t-1} + H_t^T C_{v_t}^{-1} y_t) & (2.203)\end{aligned}$$

where the notation

$$K_t = C_{\tilde{x}_{t|t-1}}H_t^T(H_t C_{\tilde{x}_{t|t-1}}H_t^T + C_{v_t})^{-1} \quad (2.204)$$

has been made. After some lengthy calculations this can be written in form [46]

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(y_t - H_t\hat{x}_{t|t-1}) \quad (2.205)$$

This is the desired recursive form; the new estimate can be updated from the prediction based on the previous one using the correction term $K_t(y_t - H_t\hat{x}_{t|t-1})$. Matrix K_t is the so-called gain matrix and the term in parenthesis is the error between the actual data value y_t and the prediction $H_t\hat{x}_{t|t-1}$.

Next we form the equation for $C_{\tilde{x}_{t|t-1}}$. First we note that

$$\begin{aligned} \tilde{x}_{t|t-1} &= x_t - \hat{x}_{t|t-1} & (2.206)\\ &= F_{t-1}x_{t-1} + G_{t-1}w_{t-1} - F_{t-1}\hat{x}_{t-1} & (2.207)\\ &= F_{t-1}\tilde{x}_{t-1} + G_{t-1}w_{t-1} & (2.208)\end{aligned}$$

and for the covariance we get

$$\begin{aligned} C_{\tilde{x}_{t|t-1}} &= E\left\{(F_{t-1}\tilde{x}_{t-1} + G_{t-1}w_{t-1})(F_{t-1}\tilde{x}_{t-1} + G_{t-1}w_{t-1})^T\right\} && (2.209)\\ &= F_{t-1}C_{\tilde{x}_{t-1}}F_{t-1}^T + G_{t-1}C_{w_{t-1}}G_{t-1}^T && (2.210)\end{aligned}$$

because $\tilde{x}_{t-1}$ and w_{t-1} are uncorrelated. For the error $\tilde{x}_t$ we have

$$\tilde{x}_t = x_t - \hat{x}_t = x_t - (\hat{x}_{t|t-1} + K_t(y_t - H_t\hat{x}_{t|t-1})) \quad (2.211)$$

Inserting $y_t = H_t x_t + v_t$ we get

$$\begin{aligned} \tilde{x}_t &= x_t - (\hat{x}_{t|t-1} + K_t(H_t x_t + v_t - H_t\hat{x}_{t|t-1})) && (2.212)\\ &= \tilde{x}_{t|t-1} - K_t(H_t\tilde{x}_{t|t-1} + v_t) && (2.213)\\ &= (I - K_tH_t)\tilde{x}_{t|t-1} - K_tv_t && (2.214)\end{aligned}$$

and for error variance

$$\begin{aligned} C_{\tilde{x}_t} &= (I - K_tH_t)C_{\tilde{x}_{t|t-1}}(I - K_tH_t)^T + K_tC_{v_t}K_t^T && (2.215)\\ &= (I - K_tH_t)C_{\tilde{x}_{t|t-1}} && (2.216)\end{aligned}$$

because $\tilde{x}_{t|t-1} = x_t - F_{t-1}\hat{x}_{t-1}$ and v_t are uncorrelated. Now equations (2.195), (2.210), (2.204), (2.216) and (2.205) form the Kalman filter. After adding the initializations we can summarize the Kalman filter algorithm

$$\begin{aligned} C_{\tilde{x}_0} &= C_{x_0} && (2.217)\\ \hat{x}_0 &= E\{x_0\} && (2.218)\\ \hat{x}_{t|t-1} &= F_{t-1}\hat{x}_{t-1} && (2.219)\\ C_{\tilde{x}_{t|t-1}} &= F_{t-1}C_{\tilde{x}_{t-1}}F_{t-1}^T + G_{t-1}C_{w_{t-1}}G_{t-1}^T && (2.220)\\ K_t &= C_{\tilde{x}_{t|t-1}}H_t^T(H_tC_{\tilde{x}_{t|t-1}}H_t^T + C_{v_t})^{-1} && (2.221)\\ C_{\tilde{x}_t} &= (I - K_tH_t)C_{\tilde{x}_{t|t-1}} && (2.222)\\ \hat{x}_t &= \hat{x}_{t|t-1} + K_t(y_t - H_t\hat{x}_{t|t-1}) && (2.223)\end{aligned}$$

CHAPTER III

Time series models

3.1 Time-varying model structures

In this chapter the time-varying data structures are closer investigated. In this context we refer with y_t to the data received at the time t. The data is thought to be an output of some time-varying dynamical system. Thus y_t has same meaning than the observation z in Chapter 2. The notations made are again due to historical reasons.

Generally the structure of the system can be expressed as the form of the nonlinear regression

$$y_t = g(t, \theta, \varphi_t) + e_t \tag{3.1}$$

where the observed output y_t is a nonlinear (time depending) function g with parameters θ of measurements φ_t. We refer to φ_t as input even if it contains – which is usual – past values of the system output y_s, $s < t$. e_t is a noise term. The objective is now to estimate the parameter vector θ based on y_s and φ_s, $s < t$. We denote this estimate with $\hat{\theta}_t$.

The goal of this chapter is to derive the most common algorithms used in estimation of the time-varying parameters of nonstationary time series.

3.2 General nonlinear recursive estimation

Let us write the model for dynamical system in form

$$y_t = \hat{y}_{t|\theta} + e_t \tag{3.2}$$

where $\hat{y}_{t|\theta} = g(t, \theta, \varphi_t)$ is a general time depending function of past data φ_t and parameter vector θ. Usually the time dependence is treated in parameters and we write

$$\hat{y}_{t|\theta} = g(\theta_t, \varphi_t) \tag{3.3}$$

The notation $\hat{y}_{t|\theta}$ emphasizes the interpretation of this quantity as a predictor of system output y_t based on parameters θ_t and data φ_t available at time t .

Let now

$$\varepsilon(t, \theta) = y_t - \hat{y}_{t|\theta} \tag{3.4}$$

where the fact that the prediction error ε is a function of the parameter θ is emphasized. Now we introduce the function

$$V_t(\theta) = \sum_{k=1}^{t} \beta(t, k)\ell(\varepsilon(k, \theta), k) \tag{3.5}$$

where ℓ is a scalar valued function. The function $V_t(\theta)$ serves as a performance index and the predictor can be estimated through the minimization of $V_t(\theta)$. A common function for ℓ is

$$\ell(x, t) = -\log p(x, t) \tag{3.6}$$

where $p(x,t)$ is the density of the noise $e(t)$. Another common choice is

$$\ell(x,t) = x^2 \tag{3.7}$$

which leads to least squares estimation. The following properties are usually required for the weighting sequence $\beta(t,k)$ [38]

$$\begin{cases} \beta(t,k) = \lambda(t)\beta(t-1,k), & 1 \le k \le t-1 \\ \beta(t,t) = 1 \end{cases} \tag{3.8}$$

and we can define the normalization parameter $\gamma(t)$

$$\gamma(t) = \left(\sum_{k=1}^{t} \beta(t,k) \right)^{-1} \tag{3.9}$$

Note that

$$\frac{1}{\gamma(t)} = \sum_{k=1}^{t-1} \beta(t,k) + \beta(t,t) = \lambda(t) \sum_{k=1}^{t-1} \beta(t-1,k) + 1 \tag{3.10}$$

$$= \frac{\lambda(t)}{\gamma(t-1)} + 1 \tag{3.11}$$

In particular, if $\lambda(t) \equiv \lambda$, then

$$\beta(t,k) = \lambda^{t-k} \tag{3.12}$$

This corresponds to exponential weighting of the fitting errors. In this case λ is called the forgetting factor and

$$\gamma(t) = \frac{1-\lambda}{1-\lambda^t} \overset{t\to\infty}{\longrightarrow} 1-\lambda \tag{3.13}$$

For recursive solution of θ typically e.g. Gauss-Newton is used.

The general algorithm for minimization of (3.5) is of the form [40]

$$\hat{\theta}_t^{(i)} = \hat{\theta}_{t-1}^{(i-1)} - \mu_t^{(i)} [R_t^{(i)}]^{-1} V_t'(\hat{\theta}_t^{(i-1)}) \tag{3.14}$$

where subscript denotes the time up to which the data has been available when the corresponding term has been established and superscript denotes the iteration counter. R_t is the search direction matrix and V_t' is the gradient of V_t with respect to parameters θ. μ_t is the iteration step size. If the iteration converges, we get the estimate $\hat{\theta}$ for model parameters θ based on data available at time t. The convergence rate of the iteration depends on the prediction error criterion V_t and the selection of μ_t and R_t.

Suppose now that for each iteration step i we observe one more measurement. In other words, when V_t is sliding over the whole data set, only one step of minimization is computed at each time instant. This leads to algorithm

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \mu_t R_t^{-1} V_t'(\hat{\theta}_t) \tag{3.15}$$

With different choices of matrix R_t different optimization methods are achieved. The simplest choice is obviously

$$R_t = I \tag{3.16}$$

which leads to gradient or steepest descent method. With

$$R_t = V_t''(\hat{\theta}_t) \tag{3.17}$$

where V_t'' is the Hessian of V_t, the resulting algorithm is called the Newton method.

It should be noted that we have not made any restrictions for function ℓ or for model structure g. For general model structure we only conclude (see [38]) without proving that if ℓ does not explicitly depend on θ the minimizing Gauss-Newton algorithm is

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma(t) R_t^{-1} \nabla_\theta(\hat{y}_{t|\theta})_t \ell'_\varepsilon(\varepsilon_t, t) \tag{3.18}$$

$$R_t = R_{t-1} + \gamma(t) \left[\nabla_\theta(\hat{y}_{t|\theta})_t \ell''_\varepsilon(\varepsilon_t, t) \nabla_\theta(\hat{y}_{t|\theta})_t^T - R_{t-1} \right] \tag{3.19}$$

If $\ell_\varepsilon(\varepsilon_t, t)$ is selected to be the least squares index $\ell(\varepsilon) = \varepsilon^2$ these equations are equal to the presentation in Section 2.10.

3.3 Recursive linear regression

Let us now make a restriction that the model is of the form

$$y_t = \varphi_t^T \theta + e_t \tag{3.20}$$

or in other words, the predictor $\hat{y}_{t|\theta}$ is the linear regression of φ_t

$$\hat{y}_{t|\theta} = \varphi_t^T \theta \tag{3.21}$$

Another restriction is that we select ℓ to be

$$\ell(x) = x^2 \tag{3.22}$$

so that

$$V_t(\theta) = \sum_{k=1}^{t} \beta(t,k)(\varepsilon(k,\theta))^2 \tag{3.23}$$

$$= \sum_{k=1}^{t} \beta(t,k)(y_t - \varphi_t^T \theta)^2 \tag{3.24}$$

is the weighted least squares criterion. This can be rewritten in form

$$V_t(\theta) = \left(y(t) - \varphi^T(t)\theta\right)^T W_t \left(y(t) - \varphi^T(t)\theta\right) \tag{3.25}$$

where

$$y(t) = (y_1, \ldots, y_t)^T \tag{3.26}$$

$$\varphi(t) = (\varphi_1, \ldots, \varphi_t) \tag{3.27}$$

and

$$W_t = \operatorname{diag}(\beta(t,1), \ldots, \beta(t,t)) \tag{3.28}$$

The estimate that minimizes this is obtained using (2.145)

$$\hat{\theta}_t = \left(\varphi(t) W_t \varphi(t)^T\right)^{-1} \varphi(t) W_t y(t) \tag{3.29}$$

$$= \overline{R}_t^{-1} f_t \tag{3.30}$$

where

$$\overline{R}_t = \sum_{k=1}^{t} \beta(t,k) \varphi_k \varphi_k^T \tag{3.31}$$

$$f_t = \sum_{k=1}^{t} \beta(t,k) \varphi_k y_k \tag{3.32}$$

Suppose that the weighting sequence has the property given (3.8). This implies that

$$\overline{R}_t = \lambda_t \overline{R}_{t-1} + \varphi_t \varphi_t^T \tag{3.33}$$
$$f_t = \lambda_t f_{t-1} + \varphi_t y_t \tag{3.34}$$

and thus we have

$$\hat{\theta}_t = \overline{R}_t^{-1} f_t = \overline{R}_t^{-1} \left[\lambda_t f_{t-1} + \varphi_t y_t\right] \tag{3.35}$$
$$= \overline{R}_t^{-1} \left[\lambda_t \overline{R}_{t-1} \hat{\theta}_{t-1} + \varphi_t y_t\right] \tag{3.36}$$
$$= \overline{R}_t^{-1} \left[\left(\overline{R}_t - \varphi_t \varphi_t^T\right) \hat{\theta}_{t-1} + \varphi_t y_t\right] \tag{3.37}$$
$$= \hat{\theta}_{t-1} + \overline{R}_t^{-1} \varphi_t \left[y_t - \varphi_t^T \hat{\theta}_{t-1}\right] \tag{3.38}$$

Usually the matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \tag{3.39}$$

is applied to inversion of $\overline{R}_t$. Introducing $P_t = \overline{R}_t^{-1}$ and taking $A = \lambda_t \overline{R}_{t-1}$, $B = D^T = \varphi_t$ and $C = 1$ gives

$$P_t = \lambda_t^{-1} \left(P_{t-1} - \frac{P_{t-1}\varphi_t \varphi_t^T P_{t-1}}{\lambda_t + \varphi_t^T P_{t-1} \varphi_t}\right) \tag{3.40}$$

Denoting $K_t = \overline{R}_t^{-1} \varphi_t$ we can summarize the Recursive Least Squares (RLS) algorithm as

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t \left[y_t - \varphi_t^T \hat{\theta}_{t-1}\right] \tag{3.41}$$
$$K_t = \frac{P_{t-1}\varphi_t}{\lambda_t + \varphi_t^T P_{t-1} \varphi_t} \tag{3.42}$$
$$P_t = \lambda_t^{-1} \left(P_{t-1} - \frac{P_{t-1}\varphi_t \varphi_t^T P_{t-1}}{\lambda_t + \varphi_t^T P_{t-1} \varphi_t}\right) \tag{3.43}$$
$$= \lambda_t^{-1} \left(I - \frac{P_{t-1}\varphi_t \varphi_t^T}{\lambda_t + \varphi_t^T P_{t-1} \varphi_t}\right) P_{t-1} \tag{3.44}$$

This result can also be obtained from the general forms (3.18) and (3.19).

3.4 State space approach to time-varying linear regression

The approach based on the weighted criterion leading to the RLS algorithm is somehow *ad hoc*. The more systematic method would be the fully Bayesian mean square approach to the recursive estimation of the parameters. For that the model for the parameter evolution is needed. For example the assumption that the parameters obey the random walk model [24] leads to system

$$\theta_t = \theta_{t-1} + w_t \tag{3.45}$$
$$y_t = \varphi_t^T \theta_t + e_t \tag{3.46}$$

This is of the form of the state space equations, and the solution $\hat{\theta}_t$ that minimizes the conditional expectation given past observations is given by Kalman filter introduced in Section 2.13.

First we make notations

$$F_t = I \tag{3.47}$$
$$G_t = I \tag{3.48}$$
$$H_t = \varphi_t^T \tag{3.49}$$
$$x_t = \theta_t \tag{3.50}$$
$$v_t = e_t \tag{3.51}$$

in (2.181) and (2.182) and we can write the Kalman filter, equations (2.217) – (2.223), in form

$$\begin{aligned}
\hat{\theta}_{t|t-1} &= \hat{\theta}_{t-1} && (3.52)\\
C_{\tilde{\theta}_{t|t-1}} &= C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}} && (3.53)\\
K_t &= C_{\tilde{\theta}_{t|t-1}}\varphi_t(\varphi_t^T C_{\tilde{\theta}_{t|t-1}}\varphi_t + C_{v_t})^{-1} && (3.54)\\
C_{\tilde{\theta}_t} &= (I - K_t\varphi_t^T)C_{\tilde{\theta}_{t|t-1}} && (3.55)\\
\hat{\theta}_t &= \hat{\theta}_{t|t-1} + K_t(y_t - \varphi_t^T\hat{\theta}_{t|t-1}) && (3.56)
\end{aligned}$$

We insert (3.52) and (3.53) into (3.54)-(3.56) and get

$$\begin{aligned}
\hat{\theta}_t &= \hat{\theta}_{t-1} + K_t(y_t - \varphi_t^T\hat{\theta}_{t-1}) && (3.57)\\
K_t &= \frac{(C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}})\varphi_t}{\varphi_t^T(C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}})\varphi_t + C_{v_t}} && (3.58)\\
C_{\tilde{\theta}_t} &= \left(I - \frac{(C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}})\varphi_t\varphi_t^T}{\varphi_t^T(C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}})\varphi_t + C_{v_t}}\right)(C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}}) && (3.59)
\end{aligned}$$

If we now denote $P_t = C_{\tilde{\theta}_t} + C_{w_t}$ the recursive mean square estimate takes the form

$$\begin{aligned}
\hat{\theta}_t &= \hat{\theta}_{t-1} + K_t(y_t - \varphi_t^T\hat{\theta}_{t-1}) && (3.60)\\
K_t &= \frac{P_{t-1}\varphi_t}{\varphi_t^T P_{t-1}\varphi_t + C_{v_t}} && (3.61)\\
P_t &= \left(I - \frac{P_{t-1}\varphi_t\varphi_t^T}{\varphi_t^T P_{t-1}\varphi_t + C_{v_t}}\right)P_{t-1} + C_{w_t} && (3.62)
\end{aligned}$$

P_t is then a recursive estimate of the covariance $C_{\tilde{\theta}_t} + C_{w_t}$ and K_t is called the Kalman gain vector. This form is seen to be very close to the form of RLS algorithm. In fact if we choose

$$\begin{aligned}
C_{v_t} &= \lambda_t && (3.63)\\
C_{w_t} &= (\lambda_t^{-1} - 1)\left(I - \frac{P_{t-1}\varphi_t\varphi_t^T}{\varphi_t^T P_{t-1}\varphi_t + C_{v_t}}\right)P_{t-1} && (3.64)
\end{aligned}$$

then the Kalman filter gives exactly the RLS algorithm. This leads to conclusion, that the RLS is optimal in mean square sense if the assumptions (3.63) and (3.64) are valid. If they cannot be done the RLS only approximates the optimal recursive estimate. The RLS approach is in practice more popular, because its performance can be tuned with one parameter λ. In fact, since λ_t is usually tuned in RLS near to unity, the implicit assumption is, that C_{w_t} is "small" corresponding to slow variation. In Kalman filter the wrong assumptions about C_{v_t} lead easily to estimates with no sense at all.

We can also note that if we choose

$$\begin{aligned}
C_{w_t} &= \mu^2\frac{\varphi_t\varphi_t^T}{\mu\varphi_t^T\varphi_t + 1} && (3.65)\\
C_{v_t} &= 1 && (3.66)\\
P_0 &= \mu I && (3.67)
\end{aligned}$$

then

$$\begin{aligned}
P_1 &= \left(I - \frac{\mu\varphi_1\varphi_1^T}{\mu\varphi_1^T\varphi_1 + 1}\right)\mu + \mu^2\frac{\varphi_1\varphi_1^T}{\mu\varphi_1^T\varphi_1 + 1} = \mu I && (3.68)\\
&\vdots\\
P_t &= \mu I && (3.69)
\end{aligned}$$

and the gain vector K_t takes the form

$$K_t = \frac{\mu\varphi_t}{\mu\varphi_t^T\varphi_t + 1} \tag{3.70}$$

The resulting algorithm is then of the form

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{\mu\varphi_t}{\mu\varphi_t^T\varphi_t + 1}\varepsilon_t \tag{3.71}$$

where

$$\varepsilon_t = y_t - \varphi_t^T\hat{\theta}_{t-1} \tag{3.72}$$

This is called the Normalized Linear Mean Square algorithm (NLMS) [24].

Yet another popular form of Kalman filter is obtained, when we select

$$\begin{aligned} C_{w_t} &= \mu^2\frac{(I-\mu\varphi_t\varphi_t^T)^{-1}\varphi_t\varphi_t^T(I-\mu\varphi_t\varphi_t^T)^{-1}}{\mu\varphi_t^T(I-\mu\varphi_t\varphi_t^T)^{-1}\varphi_t+1} - \mu(I-\mu\varphi_t\varphi_t^T)^{-1} \\ &\quad +\mu(I-\mu\varphi_{t+1}\varphi_{t+1}^T)^{-1} && (3.73)\\ C_{v_t} &= 1 && (3.74)\\ P_0 &= \mu(I-\mu\varphi_1\varphi_1^T)^{-1} && (3.75) \end{aligned}$$

Using these we get the covariance

$$P_t = \mu(I-\mu\varphi_{t+1}\varphi_{t+1}^T)^{-1} \tag{3.76}$$

and the Kalman gain vector

$$K_t = \frac{\mu(I-\mu\varphi_t\varphi_t^T)^{-1}\varphi_t}{\mu\varphi_t^T(I-\mu\varphi_t\varphi_t^T)^{-1}\varphi_t+1} \tag{3.77}$$

Using matrix inversion lemma, we can write

$$(I-\mu\varphi_t\varphi_t^T)^{-1} = I - \frac{\mu\varphi_t\varphi_t^T}{\mu\varphi_t^T\varphi_t - 1} \tag{3.78}$$

and thus the Kalman gain vector can be written in form

$$\begin{aligned} K_t &= \frac{\mu\left(I-\frac{\mu\varphi_t\varphi_t^T}{\mu\varphi_t^T\varphi_t-1}\right)\varphi_t}{\mu\varphi_t^T\left(I-\frac{\mu\varphi_t\varphi_t^T}{\mu\varphi_t^T\varphi_t-1}\right)\varphi_t+1} && (3.79)\\ &= \frac{\mu\varphi_t(\mu\varphi_t^T\varphi_t-1)-\mu^2(\varphi_t^T\varphi_t)\varphi_t}{\mu(\varphi_t^T\varphi_t)\mu(\varphi_t^T\varphi_t)-\mu(\varphi_t^T\varphi_t)-\mu^2(\varphi_t^T\varphi_t)^2+\mu\varphi_t^T\varphi_t-1} && (3.80)\\ &= -\mu^2(\varphi_t^T\varphi_t)\varphi_t+\mu\varphi_t+\mu^2(\varphi_t^T\varphi_t)\varphi_t && (3.81)\\ &= \mu\varphi_t && (3.82) \end{aligned}$$

This leads to algorithm

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \mu\varphi_t\varepsilon_t \tag{3.83}$$

Which is called the Least Mean Square (LMS) algorithm [24].

All the proposed methods in this chapter are based on the recursion

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t(y_t - \hat{\varphi}_t^T\hat{\theta}_{t-1}) \tag{3.84}$$

Only the choice of the gain K_t depends on the method. The choices for methods introduced here are summarized in Table 3.1.

TABLE 3.1
Kalman gain K_t and covariance estimate P_t for different adaptive algorithms

	K_t	P_t
Kalman	$P_{t-1}\varphi_t(\varphi_t^T P_{t-1}\varphi_t + C_{v_t})^{-1}$	$\left(I - K_t\varphi_t^T\right) P_{t-1} + C_{w_t}$
RLS	$P_{t-1}\varphi_t(\varphi_t^T P_{t-1}\varphi_t + \lambda)^{-1}$	$\lambda^{-1}\left(I - K_t\varphi_t^T\right) P_{t-1}$
NLMS	$\mu\varphi_t(\mu\varphi_t^T\varphi_t + \lambda)^{-1}$	μI
LMS	$\mu\varphi_t$	$\mu(I - \mu\varphi_{t+1}\varphi_{t+1}^T)^{-1}$

3.5 Time series modeling

3.5.1 Stationary models

As introduced in Chapter 1 one approach to time series modeling is to model the measured signal as output of linear system. If the signal is stationary, this system can be time-invariant. For example if we want to model the signal y_t with a p'th order AR model, the observation model is of the form

$$y_t = \sum_{k=1}^{p} a_k y_{t-k} + e_t \quad t = p+1, \ldots, N \tag{3.85}$$

This can be written in matrix form

$$Y = H\theta + e \tag{3.86}$$

where $\theta = (a_1, \ldots, a_p)^T$ and

$$H = \begin{pmatrix} y_p & \cdots & y_1 \\ \vdots & & \vdots \\ y_{N-1} & \cdots & y_{N-p} \end{pmatrix} \tag{3.87}$$

$$Y = (y_{p+1}, \ldots, y_N)^T \tag{3.88}$$

$$e = (e_{p+1}, \ldots, e_N)^T \tag{3.89}$$

The least squares solution to the AR parameters is then

$$\hat{\theta}_{\mathrm{LS}} = (H^T H)^{-1} H^T Y \tag{3.90}$$

$\hat{\theta}_{\mathrm{LS}}$ is the solution which minimizes the output least squares error norm

$$\hat{\theta}_{\mathrm{LS}} = \arg\min_{\theta} \|e\|^2 \tag{3.91}$$

$$= \arg\min_{\theta} e^T e \tag{3.92}$$

$$= \arg\min_{\theta} \|Y - H\theta\|^2 \tag{3.93}$$

The LS estimate of the AR model parameters is thus a linear function of data. This is not the case with ARMA modeling. In ARMA modeling the observation model can be written in form

$$y_t = \sum_{j=1}^{p} a_j y_{t-j} + \sum_{k=1}^{q} b_k e_{t-k} + e_t \tag{3.94}$$

The LS solution for parameters a and b is then the one that minimizes the norm of the error e. This minimization cannot be written in form

$$\hat{\theta}_{\mathrm{LS}} = \arg\min_{\theta} \|Y - H\theta\|^2 \tag{3.95}$$

where H is a known deterministic matrix. Therefore the ARMA model has to be calculated as nonlinear minimization problem. For different methods for solving ARMA(p, q) model in case of stationary signals see e.g. [8]. General references for estimation of stationary models are e.g. [45, 7, 35].

3.5.2 Time dependent modeling

If the signal to be modeled is nonstationary it cannot be modeled as output of a time-invariant system. It is natural to assume in this case that the system has time-varying parameters. For example the time-varying ARMA model can be written in form

$$y_t = \sum_{j=1}^{p} a_j(t) y_{t-j} + \sum_{k=1}^{q} b_k(t) e_{t-k} + e_t \tag{3.96}$$

If we denote

$$\theta_t = (a_1(t), \ldots, a_p(t), b_1(t), \ldots, b_q(t))^T \tag{3.97}$$
$$\varphi_t = (y_{t-1}, \ldots, y_{t-p}, e_{t-1}, \ldots, e_{t-q})^T \tag{3.98}$$

we can write the equation in the form

$$y_t = \varphi_t^T \theta_t + e_t \tag{3.99}$$

This is exactly of the form of linear regression (3.20). The parameters θ_t can be now estimated iteratively for example with RLS algorithm.

$$\hat{\varphi}_t = (y_{t-1}, \ldots, y_{t-p}, \varepsilon_{t-1}, \ldots, \varepsilon_{t-q})^T \tag{3.100}$$
$$\varepsilon_t = y_t - \hat{\varphi}_t^T \hat{\theta}_{t-1} \tag{3.101}$$
$$K_t = P_{t-1} \frac{\hat{\varphi}_t}{\lambda + \hat{\varphi}_t^T P_{t-1} \hat{\varphi}_t} \tag{3.102}$$
$$\hat{\theta}_t = \hat{\theta}_{t-1} + \varepsilon_t K_t \tag{3.103}$$
$$P_t = \lambda^{-1}(I - K_t \hat{\varphi}_t^T) P_{t-1} \tag{3.104}$$

where

$$\hat{\theta}_t = (\hat{a}_1(t), \ldots, \hat{a}_p(t), \hat{b}_1(t), \ldots, \hat{b}_q(t))^T \tag{3.105}$$

Note that the unknown process e_t is here estimated with the prediction error process ε_t in every step of the iteration. We use therefore the notation $\hat{\varphi}_t$ instead of φ_t.

The application to AR and MA models is straightforward. Selection of regressor and parameter vector structure for different model structures is summarized in Table 3.2.

Rest of the recursive estimators studied in this chapter are applied to time-varying time-series modeling analogously. The recursion

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t(y_t - \hat{\varphi}_t^T \hat{\theta}_{t-1}) \tag{3.106}$$

in combination of tables 3.2 and 3.1 summarizes the different combinations.

The time-varying algorithms presented here are all in their generic forms. In many cases the effectiveness of the algorithm can be tuned with different matrix decompositions and scalings during the iteration. Also the form of the linear system can be different from the linear regression used here. AR, ARMA and MA models are all so called transversal models for time series. Another

TABLE 3.2

Regressor vector $\hat{\varphi}_t$ and parameter vector $\hat{\theta}$ for different time series model structures

	$\hat{\varphi}_t^T$	$\hat{\theta}^T$
AR	$(y_{t-1}, \ldots, y_{t-p})$	$(\hat{a}_1(t), \ldots, \hat{a}_p(t))$
ARMA	$(y_{t-1}, \ldots, y_{t-p}, \varepsilon_{t-1}, \ldots, \varepsilon_{t-q})$	$(\hat{a}_1(t), \ldots, \hat{a}_p(t), \hat{b}_1(t), \ldots, \hat{b}_q(t))$
MA	$(\varepsilon_{t-1}, \ldots, \varepsilon_{t-q})$	$(\hat{b}_1(t), \ldots, \hat{b}_q(t))$

possibility is to use so called lattice structures [43]. Also for these it is possible to form recursive forms [12].

A review of adaptive algorithms is e.q. [24]. Another common reference to recursive systems is [41]. The connection of the different computational forms of the adaptive algorithms with the Kalman filter is summarized in [57]. For historical reasons also the references [67] and [26] are worth to be mentioned. The preprint books [59] and [62] contain many of the first papers published on subject of adaptive and Kalman filtering.

CHAPTER IV

Root tracking algorithms

In (stochastic) time-varying algorithms for time series modeling the parameter vector is updated during iteration with equation

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \Delta\hat{\theta}_t \tag{4.1}$$

where $\hat{\theta}_t$ is the vector containing the estimated parameters of the time series at the time t and $\Delta\hat{\theta}_t$ is the parameter update between times $t-1$ and t. That is the case in e.g. time-varying linear regression. In the case of a linear transversal model for time series, the model parameters are seldom of particular interest themselves. Usually they do not correspond to any signal characteristic separately. Often more informative is for example the spectrum of the signal. As seen from equation (1.27) the spectrum is a function of the model parameters. The estimate for the momentary spectrum of the signal can then be obtained using the estimated time-varying parameters of the signal. In [36] the term instantaneous spectrum has been used. Thus we can *define* the momentary spectrum as

$$S_t(\omega) = \sigma_e^2 \frac{\left|1 + B_t(e^{i\omega})\right|^2}{\left|A_t(e^{i\omega})\right|^2} \tag{4.2}$$

The equation of the spectrum can clearly be written in the form

$$S(\omega) = \sigma_e^2 \frac{\left|\prod_{k=1}^{q}(1 - \xi_k e^{-i\omega})\right|^2}{\left|\prod_{j=1}^{p}(1 - \zeta_j e^{-i\omega})\right|^2} \tag{4.3}$$

where ξ_k and ζ_j are the roots of the operator polynomials $(1 - B(q))$ and $A(q)$ respectively [21].

When the signal characteristics are to be tracked, the time variation of the spectral peaks are in many cases of the particular interest. As seen from equation (4.3), the peaks in spectrum have strong connection to the roots of the denominator, that is, the roots of the polynomial $A(q)$. Thus the roots contain information about the spectral characteristics of the signal themselves and it is possible to make inference about the spectral content of the signal based on the roots of $A(q)$. A typical example is the detection of narrow band waveforms from signal. One possibility for solving the roots of a polynomial having time-varying coefficients is to use an adaptive root tracking method.

Examples of situations, where the polynomial roots have been considered more appropriate than the polynomial coefficients, include direction of arrival estimation [58], frequency tracking of multiple narrow band signals [11], detection of ventricular fibrillation [22], some EEG modeling problems [56, 51, 52] and estimation of heart rate variability [42].

Several approaches have been used for adaptive root tracking. The most obvious is to use the Newton's method, which is a fixed point type iteration to solve one root of polynomial [51]. Newton's iteration applied to simultaneous solution of a complex pair of roots is called the Bairstow's method. The method has been used in [31] for adaptive solution of all roots of polynomial. First order Taylor approximation of the coefficients to roots mapping has been used in [49]. In all these

methods the approach is to approximate the roots of the polynomial from precalculated polynomial coefficients. In [48, 64] a totally different approach has been adopted. The recursive least squares solution has been proposed directly to parameter vector containing the roots of the system polynomials. A further class of approaches that has been used in root tracking are the homotopy continuation methods [1, 63].

In this chapter some of these methods are introduced in detail.

4.1 Newton's method

In Section 2.10 we proposed recursive methods for minimization of a quadratic function. The minimization was done by finding the zero of the derivative of the function to be minimized. In this chapter we are interested in more generally to find a zero of a nonlinear function. Of special interest is finding of single roots of a polynomial.

Suppose that $F : \mathbb{C} \to \mathbb{C}$ is a differentiable scalar function of a scalar variable q. Let $F(q)$ has a zero ζ^* such that

$$F(\zeta^*) = 0 \tag{4.4}$$

Let ζ_i be an estimate for ζ^*. Then the first order Taylor's expansion for $F(\zeta_{i+1})$ is

$$F(\zeta_{i+1}) \approx F(\zeta_i) + F'(\zeta_i)(\zeta_{i+1} - \zeta_i) \tag{4.5}$$

Using this as an approximation for $F(\zeta^*)$ we get

$$F(\zeta_i) + F'(\zeta_i)(\zeta_{i+1} - \zeta_i) \approx 0 \tag{4.6}$$

Solving for ζ_{i+1}, we obtain

$$\zeta_{i+1} = \zeta_i - \frac{F(\zeta_i)}{F'(\zeta_i)} \tag{4.7}$$

Further as in Section 2.10 it is reasonable to introduce a step size parameter α that controls the search distance. This recursion is the Newton's method for finding a root of the algebraic equation. The convergence of the Newton's method is a well understood question. For results for this see e.g. [34].

In this section we are in particular interested in finding a single root of the polynomial $A(q) = a_0 + \ldots + a_p q^{-p}$. For simplicity, we write the method rather for polynomial $q^p A(q)$ which has the same roots as $A(q)$ with expectation of p-time zero in $q = 0$. Thus the algebraic equation is of the form

$$q^p A(q) = 0 \tag{4.8}$$

with solution $q = \zeta^*$. The derivation of $q^p A(q)$ gives

$$\begin{aligned} D(q^p A(q)) &= D(a_0 q^p + \ldots + a_p) && (4.9)\\ &= p a_0 q^{p-1} + \ldots + a_{p-1} && (4.10) \end{aligned}$$

and the recursion (4.7) can be written in form

$$\begin{aligned} \zeta_{i+1} &= \zeta_i - \alpha \frac{\zeta_i^p A(\zeta_i)}{D(q^p A(q))(\zeta_i)} && (4.11)\\ &= \zeta_i - \alpha \frac{a_0 \zeta_i^p + \ldots + a_p}{p a_0 \zeta_i^{p-1} + \ldots + a_{p-1}} && (4.12) \end{aligned}$$

where α is the step size parameter. When using complex arithmetic and a complex initial value for ζ_0 this iteration can converge to a complex root of polynomial $A(q)$. The convergence to the desired root has to be guaranteed by choosing the initial value close enough to the desired root.

The use of the Newton's method for tracking of the root of a polynomial is straightforward. The polynomial coefficients are then treated as time-varying coefficients, and one step of the iteration is calculated. The estimated root of the preceding time instant is used as initial root.

In practice it is recommended, that the values of the polynomial and the derivative are calculated with the algorithm called Horner's rule. For implementation and performance of this see [25].

Note that in this method all the polynomials can be calculated before applying the root tracking step to root estimation. In fact this method can be seen as nonlinear recursive filter with time-varying filter coefficients applied to the vector valued process of the model parameters.

4.2 First order approximation

In this approach the function that maps the polynomial coefficients to the roots is approximated with the first order Taylor approximation. The approximated roots can then be used in combination with ordinary adaptive methods for time-varying signals.

First we note, that the function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be differentiable at c if there exists linear $T_c : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$f(c+v) = f(c) + T_c(v) + \|v\| E_c(v) \tag{4.13}$$

where $E_c(v) \to 0$ as $\|v\| \to 0$. This is called the first order Taylor formula [2]. It is well known that

$$T_c(u) = f'(c;u) \tag{4.14}$$

where $f'(c;u)$ is the directional derivative of $f(c)$ in direction u, and

$$f(c+v) = f(c) + f'(c;v) + \|v\| E_c(v) \tag{4.15}$$

Now let $A_t(q)$ be a polynomial of order p. As polynomial it can be written in form

$$A_t(q) = a_0(t) + \cdots + a_p(t)q^{-p} = a_0(t)\left(1 - \zeta_1(t)q^{-1}\right)\cdots\left(1 - \zeta_p(t)q^{-1}\right) \tag{4.16}$$

We denote the vector of the coefficients and vector of the roots by $a(t)$ and $\zeta(t)$ respectively. Assume that $\zeta(t) = \zeta(a(t))$ is a $\mathbb{R}^{p+1} \to \mathbb{R}^p$ function which maps the polynomial coefficients to its roots. The problem is now to approximate the polynomial roots when the coefficient vector is updated by

$$a(t+1) = a(t) + \Delta a(t) \tag{4.17}$$

The root update equation

$$\zeta(t+1) = \zeta(t) + \Delta\zeta(t) \tag{4.18}$$

is then

$$\zeta(a(t+1)) = \zeta(a(t) + \Delta a(t)) \tag{4.19}$$

From the first order Taylor equation we get

$$\zeta(a(t) + \Delta a(t)) = \zeta(a(t)) + \zeta'(a(t))(\Delta a(t)) + \|\Delta a(t)\| E(\Delta a(t)) \tag{4.20}$$

so that the approximated update for root vector is of the form

$$\Delta\zeta(t) \approx \zeta'(a(t))(\Delta a(t)) \tag{4.21}$$

From (4.16) we get for $\zeta_i(t)$

$$\zeta_i(t) = \frac{-\sum_{j=0}^{p} a_j(t)q^{-j}}{a_0(t)\prod_{\substack{j=1\\ j\neq i}}^{p}(1 - \zeta_j(t)q^{-1})}q + q \tag{4.22}$$

and assuming the zeros are simple, partial derivation with respect to $a_m(t)$ gives

$$\frac{\partial\zeta_i(t)}{\partial a_m(t)} = \frac{-q^{-m}}{a_0(t)\prod_{\substack{j=1\\ j\neq i}}^{p}(1 - \zeta_j(t)q^{-1})}q \tag{4.23}$$

and when this is evaluated at $q = \zeta_i(t)$, we get

$$\left.\frac{\partial \zeta_i(t)}{\partial a_m(t)}\right|_{q=\zeta_i(t)} = \frac{-\zeta_i(t)^{-m+1}}{a_0(t)\prod_{\substack{j=1\\ j\neq i}}^{p}\left(1-\zeta_j(t)\zeta_i^{-1}(t)\right)} \tag{4.24}$$

$$= \frac{-\zeta_i(t)^{p-m}}{a_0(t)\prod_{\substack{j=1\\ j\neq i}}^{p}\left(\zeta_i(t)-\zeta_j(t)\right)} \tag{4.25}$$

The derivative of the function ζ is then similar to the transformation matrix

$$\zeta_a'(a(t)) = \begin{pmatrix} \frac{\partial \zeta_1(t)}{\partial a_0(t)} & \cdots & \frac{\partial \zeta_1(t)}{\partial a_p(t)} \\ \vdots & & \vdots \\ \frac{\partial \zeta_p(t)}{\partial a_0(t)} & \cdots & \frac{\partial \zeta_p(t)}{\partial a_p(t)} \end{pmatrix}_{\zeta=\zeta(t)} \tag{4.26}$$

and the root update is calculated by matrix vector multiplication

$$\Delta\zeta(t) \approx \zeta_a'(a(t))\Delta a(t) \tag{4.27}$$

or componentwise as the sum

$$\Delta\zeta_i(t) = \sum_{m=0}^{p} \frac{\partial \zeta_i(t)}{\partial a_m(t)}\Delta a_m(t) \tag{4.28}$$

The implementation of the method for time-varying estimators is straightforward. For example the LMS algorithm for AR models can be written in form

$$\hat{\varphi}_t = (y_{t-1},\ldots,y_{t-p})^T \tag{4.29}$$

$$\varepsilon_t = y_t - \hat{\varphi}_t^T\hat{\theta}_{t-1} \tag{4.30}$$

$$\Delta\hat{\theta}_{t-1} = \mu\varphi_t\varepsilon_t \tag{4.31}$$

$$\Delta\zeta(t-1) = \zeta_\theta'(\hat{\theta}_{t-1})\Delta\hat{\theta}_{t-1} \tag{4.32}$$

$$\zeta(t) = \zeta(t-1) + \Delta\zeta(t-1) \tag{4.33}$$

$$\hat{\theta}_t = \Phi(\zeta(t)) \tag{4.34}$$

$$\tag{4.35}$$

where

$$\hat{\theta}_t = (\hat{a}_1(t),\ldots,\hat{a}_p(t))^T \tag{4.36}$$

and Φ is the inverse mapping for ζ, that is $\Phi(\zeta(t))$ maps the polynomial roots to polynomial coefficients. The RLS case is treated identically

$$\hat{\varphi}_t = (y_{t-1},\ldots,y_{t-p})^T \tag{4.37}$$

$$\varepsilon_t = y_t - \hat{\varphi}_t^T\hat{\theta}_{t-1} \tag{4.38}$$

$$K_t = P_{t-1}\frac{\hat{\varphi}_t}{\lambda + \hat{\varphi}_t^T P_{t-1}\hat{\varphi}_t} \tag{4.39}$$

$$\Delta\hat{\theta}_{t-1} = K_t\varepsilon_t \tag{4.40}$$

$$\Delta\zeta(t-1) = \zeta_\theta'(\hat{\theta}_{t-1})\Delta\hat{\theta}_{t-1} \tag{4.41}$$

$$\zeta(t) = \zeta(t-1) + \Delta\zeta(t-1) \tag{4.42}$$

$$\hat{\theta}_t = \Phi(\zeta(t)) \tag{4.43}$$

$$P_t = \lambda^{-1}(I - K_t\hat{\varphi}_t^T)P_{t-1} \tag{4.44}$$

In the ARMA case both of the polynomials can be treated separately.

4.3 Direct root estimation

One approach is to estimate the roots of system polynomials without explicitely estimating the polynomial coefficients [48]. The polynomials can be parametrized directly in terms of polynomial roots. The roots can be presented either in the cartesian or in the polar form. In the polar form the system parameters are then the moduli and the angles of the roots of the polynomials. The prediction error can be now written as a function of these system parameters, and the formulation of a recursive estimate can be done with use of theory explained in Section 3.2.

We discuss here only the AR case. As introduced in Chapter 1, the AR(p) model for a process can be written in form

$$A(q)y_t = e_t \tag{4.45}$$

The polynomial can be presented in the sum form or in the product form

$$A(q) = 1 - \sum_{j=1}^{p} a_j q^{-j} = \prod_{j=1}^{p} (1 - \zeta_j q^{-1}) \tag{4.46}$$

Note that here $a_0 \equiv 1$. The parameters ζ_j are clearly the roots of the polynomial $A(q)$. The system coefficients are assumed to be real which implies that the roots are either real or occur in complex conjugate pairs. Suppose that $A(q)$ has m pairs of complex conjugate roots and n real roots so that $p = 2m + n$. $A(q)$ can then be partitioned to second and first order sections

$$A(q) = \prod_{j=1}^{n} F_j(q) \prod_{k=1}^{m} S_k(q) \tag{4.47}$$

where

$$F_j(q) = (1 - \xi_j q^{-1}) \tag{4.48}$$
$$S_k(q) = (1 - \zeta_k q^{-1})(1 - \zeta_k^* q^{-1}) \tag{4.49}$$

where the asterisk denotes the complex conjugate. The complex roots of $S_k(q)$ can be written in form

$$\zeta_k = \rho_k e^{i\omega_k} \tag{4.50}$$
$$\zeta_k^* = \rho_k e^{-i\omega_k} \tag{4.51}$$

The parameters of the polynomial are thus

$$\theta = (\rho^T, \omega^T, \xi^T) \tag{4.52}$$

The prediction error is now

$$\varepsilon_t = y_t + \sum_{i=1}^{p} a_i y_{t-i} \tag{4.53}$$
$$= y_t - \varphi_t^T a \tag{4.54}$$

where

$$\varphi_t = (-y_{t-1}, \ldots, -y_{t-p})^T \tag{4.55}$$
$$a = (a_1, \ldots, a_p)^T \tag{4.56}$$

From (3.4) we get for output estimate

$$\hat{y}_{t|\theta} = y_t - \varepsilon_t \tag{4.57}$$

so that the gradient is

$$\nabla_\theta(\hat{y}_{t|\theta}) = -\frac{\partial \varepsilon_t}{\partial \theta^T} = -\left(\frac{\partial \varepsilon_t}{\partial \theta_1}, \ldots, \frac{\partial \varepsilon_t}{\partial \theta_p}\right) \tag{4.58}$$

This can be written in the form

$$\nabla_\theta(\hat{y}_{t|\theta}) = -\frac{\partial \varepsilon_t}{\partial a^T}\frac{\partial a}{\partial \theta^T} \tag{4.59}$$

From (4.54) we get

$$\frac{\partial \varepsilon_t}{\partial a^T} = -\varphi_t^T \tag{4.60}$$

and we form the following partitioning

$$\frac{\partial a}{\partial \theta^T} = \left(\frac{\partial a}{\partial \rho^T}, \frac{\partial a}{\partial \omega^T}, \frac{\partial a}{\partial \xi^T}\right) \tag{4.61}$$

We form the derivative $\partial a/\partial \rho^T$ first. Differentiating (4.46) and (4.47) and equalizing the resulting forms we get

$$\frac{\partial A(q)}{\partial \rho_k} = \frac{\partial S_k(q)}{\partial \rho_k}\prod_{i=1, i\neq k}^{m} S_i(q)\prod_{j=1}^{n} F_j(q) \tag{4.62}$$

$$= \sum_{i=0}^{p}\frac{\partial a_i}{\partial \rho_k}q^{-i} \tag{4.63}$$

Second order section $S_k(q)$ can be written in form

$$S_k(q) = (1-\rho_k e^{i\omega_k}q^{-1})(1-\rho_k e^{-i\omega_k}q^{-1}) \tag{4.64}$$

$$= 1-\rho_k(e^{i\omega_k}+e^{-i\omega_k})q^{-1}+\rho_k^2 q^{-2} \tag{4.65}$$

$$= 1-2\rho_k\cos(\omega_k)q^{-1}+\rho_k^2 q^{-2} \tag{4.66}$$

and thus

$$\frac{\partial S_k(q)}{\partial \rho_k} = -2\cos(\omega_k)q^{-1}+2\rho_k q^{-2} \tag{4.67}$$

Multiplying (4.63) and (4.62) by $S_k(q)$ we get

$$S_k(q)\sum_{i=0}^{p}\frac{\partial a_i}{\partial \rho_k}q^{-i} = \frac{\partial S_k(q)}{\partial \rho_k}\sum_{i=0}^{p} a_i q^{-i} \tag{4.68}$$

The derivative $\partial a_i/\partial \rho_k$ can now be calculated recursively by equating the coefficients of q^{-i}. This procedure is presented in Table 4.1.

TABLE 4.1

Coefficients of q^{-i} in left hand side (LHS) and right hand side (RHS) of equation (4.68).

i	LHS	RHS
0	$\frac{\partial a_0}{\partial \rho_k}$	0
1	$\frac{\partial a_1}{\partial \rho_k} - 2\rho_k\cos(\omega_k)\frac{\partial a_0}{\partial \rho_k}$	$-2\cos(\omega_k)a_0$
2	$\frac{\partial a_2}{\partial \rho_k} - 2\rho_k\cos(\omega_k)\frac{\partial a_1}{\partial \rho_k} + \rho_k^2\frac{\partial a_0}{\partial \rho_k}$	$2\rho_k a_0 - 2\cos(\omega_k)a_1$
$\vdots$	$\vdots$	$\vdots$
i	$\frac{\partial a_i}{\partial \rho_k} - 2\rho_k\cos(\omega_k)\frac{\partial a_{i-1}}{\partial \rho_k} + \rho_k^2\frac{\partial a_{i-2}}{\partial \rho_k}$	$2\rho_k a_{i-2} - 2\cos(\omega_k)a_{i-1}$

Thus we can summarize the recursion noting that $a_0 \equiv 1$

$$\frac{\partial a_i}{\partial \rho_k} = 2\rho_k \cos(\omega_k)\frac{\partial a_{i-1}}{\partial \rho_k} - \rho_k^2 \frac{\partial a_{i-2}}{\partial \rho_k} + 2\rho_k a_{i-2} - 2\cos(\omega_k)a_{i-1} \tag{4.69}$$

$$\frac{\partial a_0}{\partial \rho_k} = 0 \tag{4.70}$$

$$\frac{\partial a_1}{\partial \rho_k} = -2\cos(\omega_k) \tag{4.71}$$

Rest of the submatrices are found analogously. The resulting recursions are of the form

$$\frac{\partial a_i}{\partial \omega_k} = 2\rho_k \cos(\omega_k)\frac{\partial a_{i-1}}{\partial \omega_k} - \rho_k^2 \frac{\partial a_{i-2}}{\partial \omega_k} + 2\rho_k \sin(\omega_k)a_{i-1} \tag{4.72}$$

$$\frac{\partial a_0}{\partial \omega_k} = 0 \tag{4.73}$$

$$\frac{\partial a_1}{\partial \omega_k} = 2\rho_k \sin(\omega_k) \tag{4.74}$$

and

$$\frac{\partial a_i}{\partial \xi_k} = \rho_k \frac{\partial a_{i-1}}{\partial \xi_k} - a_{i-1} \tag{4.75}$$

$$\frac{\partial a_0}{\partial \xi_k} = 0 \tag{4.76}$$

The pole estimation algorithm can now be formed analogously with the Recursive Least Squares algorithm in Section 3.5.2 and with equations (3.18) and (3.19)

$$\varphi_t = (-y_{t-1}, \ldots, -y_{t-p})^T \tag{4.77}$$

$$\varepsilon_t = y_t - \varphi_t^T \hat{a}(t) \tag{4.78}$$

$$K_t = P_{t-1}\frac{\nabla_\theta(\hat{y}_{t|\theta})_t}{\lambda + \nabla_\theta(\hat{y}_{t|\theta})^T P_{t-1}\nabla_\theta(\hat{y}_{t|\theta})} \tag{4.79}$$

$$P_t = \lambda^{-1}(I - K_t \nabla_\theta(\hat{y}_{t|\theta})_t^T)P_{t-1} \tag{4.80}$$

$$\hat{a}(t) = \Phi(\hat{\theta}_t) \tag{4.81}$$

$$\frac{\partial \hat{a}(t)}{\partial \hat{\theta}_t^T} = \left[\frac{\partial a}{\partial \theta^T}\right]_{\theta=\hat{\theta}_t} \tag{4.82}$$

$$\nabla_\theta(\hat{y}_{t|\theta})_{t+1} = \frac{\partial \hat{a}(t)^T}{\partial \hat{\theta}_t}\varphi_{t+1} \tag{4.83}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \varepsilon_t K_t \tag{4.84}$$

where

$$\hat{\theta}_t = (\hat{\rho}_t^T, \hat{\omega}_t^T, \hat{\xi}_t^T)^T \tag{4.85}$$

and $\Phi(\hat{\theta}_t)$ is the function mapping the polynomial roots to the polynomial coefficients.

4.4 Bairstow's method

Bairstow's method is a method for solving the complex roots of a real polynomial [65]. It avoids complex arithmetic. Another advatage is that it does not necessitate a complex starting point for iteration. Basically the Bairstow's method is the Newton's method applied to complex pair of roots. Like Newton's method Bairstow's method can be used for tracking. Then only one step of iteration is executed at a time instant. Bairstow's method has been applied to tracking the characteristics of heart rate variability spectral parameters in [42].

First we consider a polynomial

$$A(q) = \sum_{j=0}^{p} a_j q^{-j} \tag{4.86}$$

and observe that the roots of the real quadratic polynomial

$$S(q) = 1 - uq^{-1} - vq^{-2} \tag{4.87}$$

are roots of $A(q)$ if the remainder of the fraction

$$\frac{A(q)}{S(q)} \tag{4.88}$$

vanishes. Now we write the polynomial $A(q)$ into form

$$A(q) = S(q)B(q) + [r(q-u) + s]q^{-p} \tag{4.89}$$

where $B(q)$ is a $(p-2)$'th order polynomial and $[r(q-u)+s]q^{-p}$ denotes the remainder. Comparing the right hand side and the left hand side of (4.89) we can equate the coefficients and get the order recursion

$$\begin{aligned}
a_0 &= b_0 && (4.90)\\
a_1 &= b_1 - ub_0 && (4.91)\\
a_2 &= b_2 - ub_1 - vb_0 && (4.92)\\
a_3 &= b_3 - ub_2 - vb_1 && (4.93)\\
&\vdots && (4.94)\\
a_j &= b_j - ub_{j-1} - vb_{j-2} && (4.95)\\
&\vdots && (4.96)\\
a_{p-1} &= r - ub_{p-2} - vb_{p-3} && (4.97)\\
a_p &= s - ru - vb_{p-2} && (4.98)
\end{aligned}$$

and from the last two equations we can write for the terms of the remainder

$$\begin{aligned}
r(u,v) &= a_{p-1} + ub_{p-2} + vb_{p-3} && (4.99)\\
s(u,v) &= a_p + ru + vb_{p-2} && (4.100)
\end{aligned}$$

We can now apply the Newton's method to find the solution to $r(u,v) = s(u,v) = 0$. This leads to recursion

$$\begin{pmatrix} u_{i+1} \\ v_{i+1} \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix} - J^{-1}(u_i, v_i) f(u_i, v_i) \tag{4.101}$$

where

$$f(u_i, v_i) = \begin{pmatrix} r(u_i, v_i) \\ s(u_i, v_i) \end{pmatrix} \tag{4.102}$$

and

$$J(u_i, v_i) = \begin{pmatrix} \frac{\partial r}{\partial u} & \frac{\partial r}{\partial v} \\ \frac{\partial s}{\partial u} & \frac{\partial s}{\partial v} \end{pmatrix}_{\substack{u=u_i \\ v=v_i}} \tag{4.103}$$

For the partial derivatives we get

$$\frac{\partial r}{\partial u} = b_{p-2} \tag{4.104}$$

$$\frac{\partial r}{\partial v} = b_{p-3} \tag{4.105}$$

$$\frac{\partial s}{\partial u} = \frac{\partial r}{\partial u}u + r = b_{p-2}u + r \tag{4.106}$$

$$\frac{\partial s}{\partial v} = \frac{\partial r}{\partial v}u + b_{p-2} = b_{p-3}u + b_{p-2} \tag{4.107}$$

and the inverse of the Jacobian is easy to calculate, because it is a 2×2 matrix. The algorithm thus calculates iteratively the coefficients u and v of the polynomial $S(q)$. The roots of this can be easily calculated in closed form.

CHAPTER V

Simulations

An adaptive estimator has in general two types of errors. The first can be seen identical to variance of any estimator. Variance of an adaptive estimator is dependent on the memory (size of the data window) of the algorithm in the same manner than the variance of the nonrecursive estimator depends on the sample size. The longer the memory is the smaller the steady state variance is. If the memory is infinite, the variance should tend to zero if the estimator is consistent and the data is stationary. Another type of error is the so-called lag misadjustment. This can be seen as the momentary bias of the estimator.

Both types of the error have to be notified when the tracking ability of an algorithm is investigated. The tracking ability is usually a compromise between the variance and the momentary bias.

The tracking ability of an adaptive algorithm is not easily evaluated. Usually this necessitates implicit assumptions about the true parameter evolutions. For a review see [39]. The most common algorithm studied is the RLS algorithm due to its easily understood properties [18].

The performance analysis of an adaptive algorithm does not get easier when a root tracking part is added to it. In general, the convergence of any root calculation algorithm is a function of all the roots of polynomial. Thus the performance of the tracking algorithm depends strongly on the location of all the roots on the complex plane. When modeling a time series especially in case of overmodeling, the variances of the roots can be large. The performance of the root tracking algorithm can thus vary very much during iteration.

Also the rate of the root movement is of great importance. If the rate is too high some root tracking algorithms can loose the track. In other words they start to track some other root of the time-varying polynomial. Some algorithms like the complex Newton's algorithm must be re-initialized if they start to follow real root of the polynomial.

These are the reasons why the performance analysis is not straightforward even with simulations. In simulations we have to make compromises in type and order of the processes and of the models used. We also have to restrict the root movements and locations to some limited area in the complex plane.

5.1 Simulation of the processes

Due to facts given above we have made the following choices for simulations:

1. All the algorithms are evaluated with RLS based choice for the parameter update direction.
2. The processes are realizations of time-varying AR(2) processes.
3. Only AR(4) models are used for estimation.

With these choices we avoid the complexity arising from the "dummy" roots.

The true root evolutions were generated with equation

$$\zeta(t) = \rho(t)e^{i\omega(t)} \tag{5.1}$$

where

$$\rho(t) = \alpha_1 \sin(\alpha_2 t) + \alpha_3 \tag{5.2}$$
$$\omega(t) = \beta_1 \cos(\beta_2 t) + \beta_3 \tag{5.3}$$

With $\alpha_2 = \beta_2$ the trajectory of the root is thus a "bended" ellipsoid with the center point (α_3, β_3). (α_1, β_1) controls the size of the ellipsoid and (α_2, β_2) the rate of the change of the root. Typical root evolution is shown in Fig. 5.1 with parameters

$$(\alpha_1, \beta_1) = (0.03, \pi/9) \tag{5.4}$$
$$(\alpha_2, \beta_2) = (0.005\pi, 0.005\pi) \tag{5.5}$$
$$(\alpha_3, \beta_3) = (0.9, \pi/4) \tag{5.6}$$

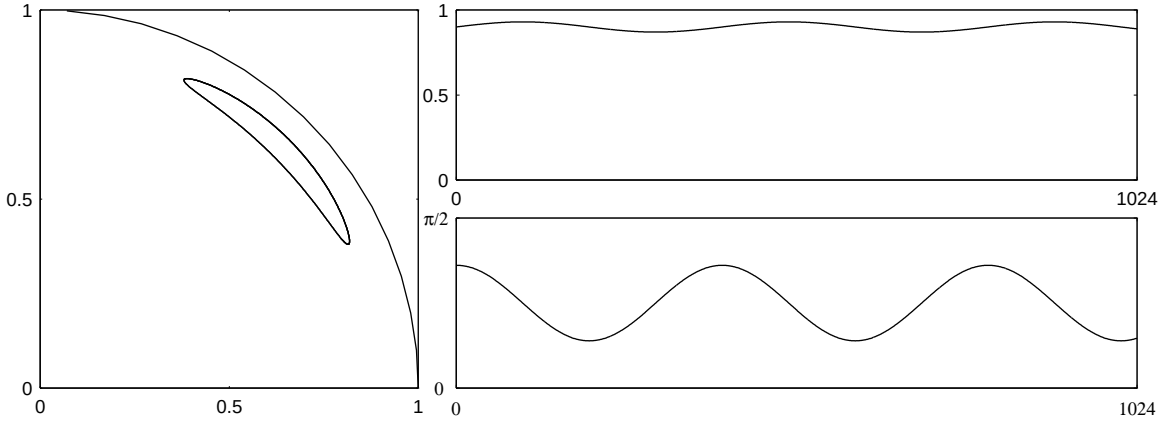

Figure 5.1: A typical example of a root evolution. Complex trajectory (left), magnitude (top) and angle (bottom).

The time-varying second order polynomial was then formed with equation

$$A(q)(t) = (1 - \zeta(t))(1 - \zeta^*(t)) = 1 - a_1(t)q^{-1} - a_2(t)q^{-2} \tag{5.7}$$

and the time-varying AR(2) process was generated using equation

$$y_t = \sum_{j=1}^{2} a_j(t) y_{t-j} + e_t \tag{5.8}$$

where e_t is a Gaussian white noise process. Theoretical spectrum of the TVAR(2) process corresponding to the root evolution of Fig. 5.1 is shown in Fig. 5.2. A typical realization is shown in Fig. 5.3.

The spectrogram calculated with FFT is shown in Fig. 5.4 using reasonable length and overlapping of the data window.

5.2 Evaluation of methods

For the evaluation of the Newton's method the TVAR(4) model was calculated with the RLS algorithm for the realization shown in Fig. 5.3. Forgetting factor $\lambda = 0.95$ was used. The momentary spectrum of the resulting TVAR model is shown in Fig. 5.5.

The Newton's mehtod was then applied to estimation of the roots. Several step size factors α were evaluated visually, and the result of the best estimate is shown in Fig. 5.6. The same was

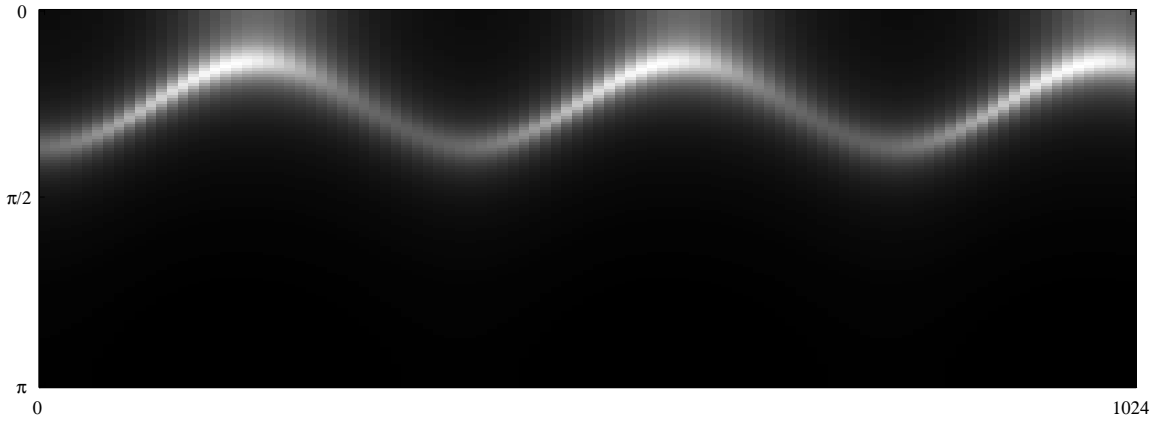

Figure 5.2: Theoretical spectrum of the time-varying AR(2) process with parameters (5.4) – (5.6).

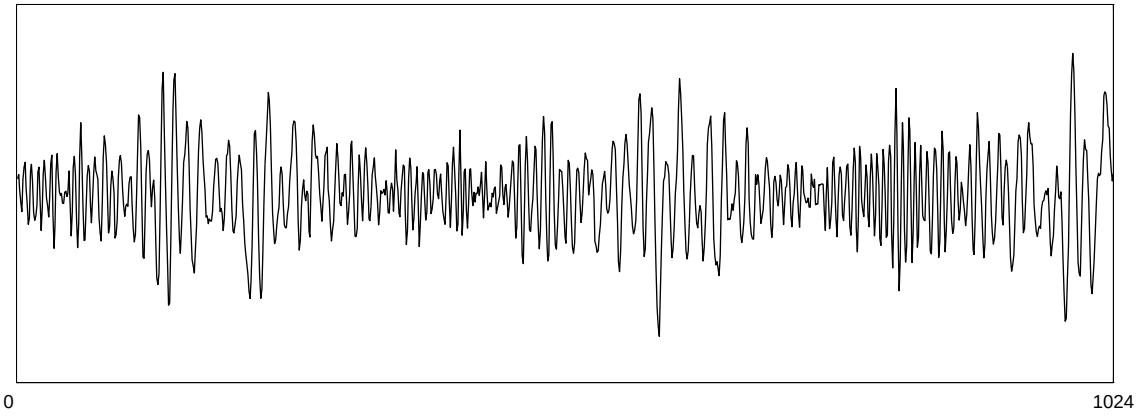

Figure 5.3: Typical realization of the time-varying AR(2) process with parameters (5.4) – (5.6).

repeated with the first order approximation method using $p = 4$ and $\lambda = 0.95$. The result is shown in Fig. 5.7.

The two methods are compared in Fig. 5.8 and Fig. 5.9. The lag error is clearly identifiable in both of the methods. The Newton's method has somewhat smaller error in this test. The α value used here for the first order approximation method was the largest with which the method had no stability problems. When the rate of the root movement increases, the stability will be even bigger problem than in this studied case.

The two other algorithms presented in previous chapter had serious stability problems during simulations.

Next we simulated the performance of the Newton's method with a rapidly changing process. We created five length $N = 256$ realizations of the process with parameters of the roots

$$(\alpha_1, \beta_1) = (0.03, \pi/9) \tag{5.9}$$

$$(\alpha_2, \beta_2) = (0.02\pi, 0.02\pi) \tag{5.10}$$

$$(\alpha_3, \beta_3) = (0.9, \pi/4) \tag{5.11}$$

The roots are shown in Fig. 5.10

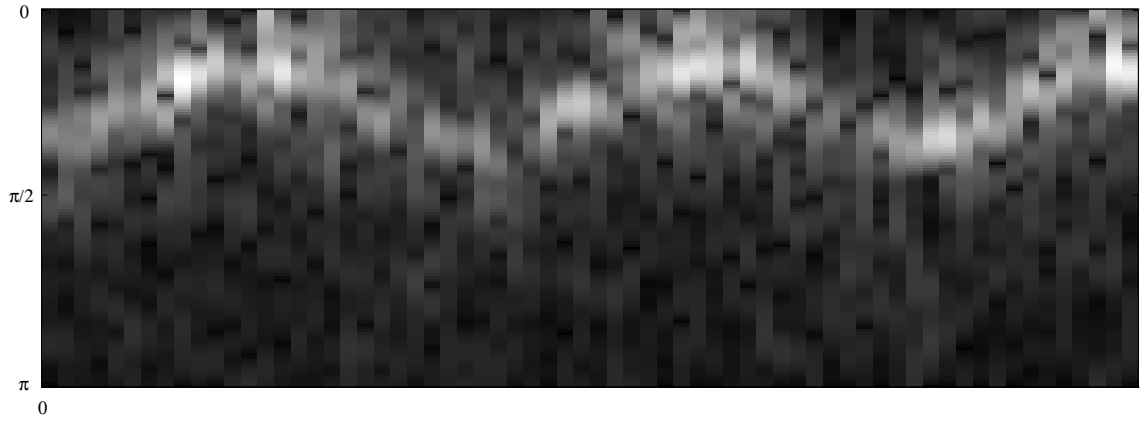

Figure 5.4: The spectrogram of the realization shown in 5.3.

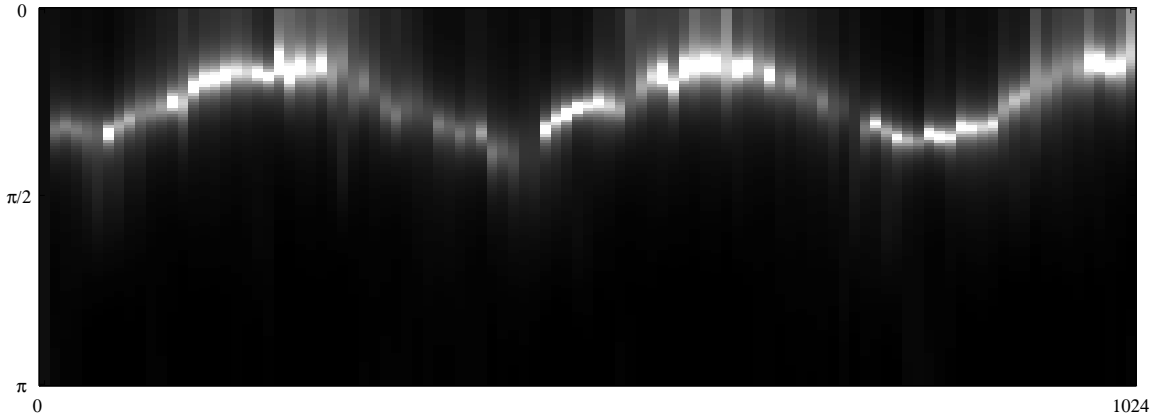

Figure 5.5: The time varying spectrum of the AR(4) model calculated from the realization shown in Fig. 5.3 using the RLS algorithm.

The TVAR(4) model was then calculated with different values of λ for each of the realizations. For each of the parameter processes, the roots were calculated with the Newton's method with different values of α. The norm of the difference was then calculated for the true and the approximated roots. The errors of the five realizations were then added together. The contour plot of the error is shown in Fig. 5.11.

It is seen that the error is in minimum around the $(\alpha, \lambda) \approx (0.25, 0.8)$. The error is even smaller than with larger values of α which are closer to the true roots of the parameter vector. This is due to the filtering effect of the Newton's method.

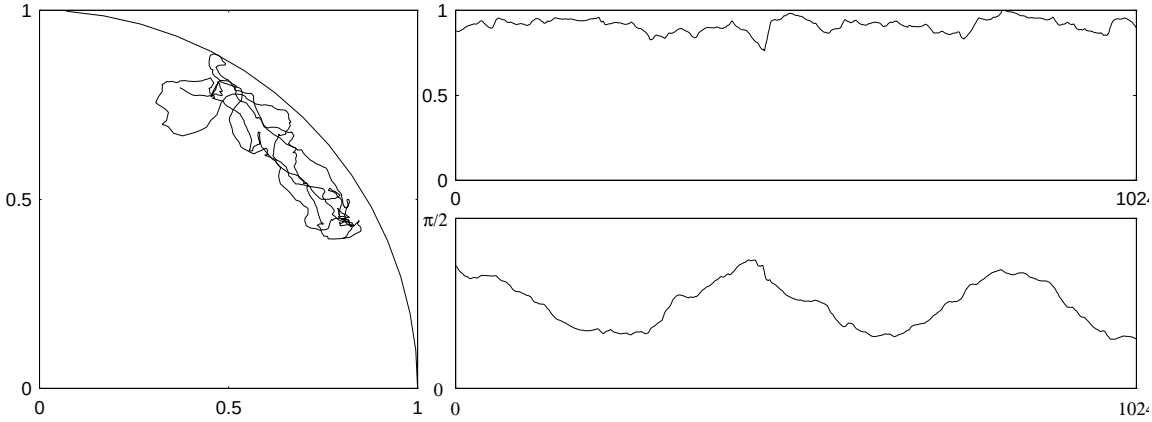

Figure 5.6: The roots estimated with the Newton's method. $\alpha = 0.1$

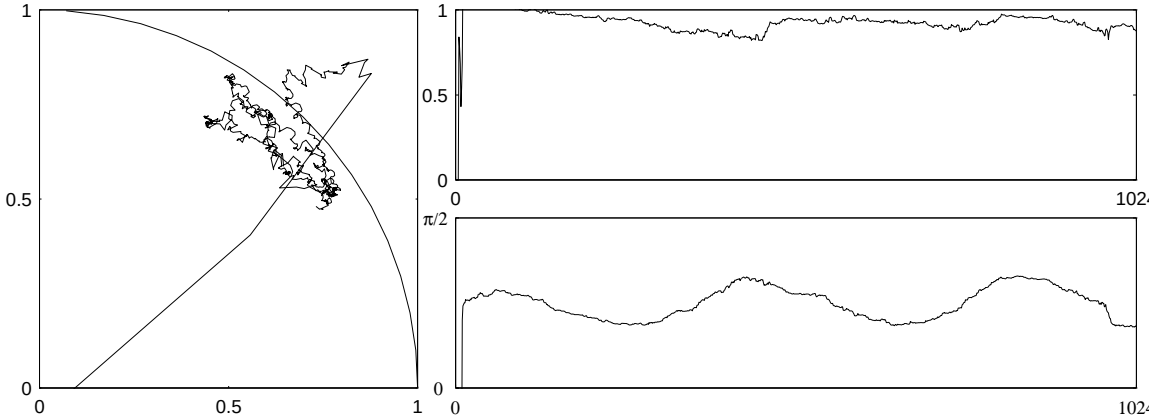

Figure 5.7: The roots estimated with the first order approximation method. $\alpha = 0.3$

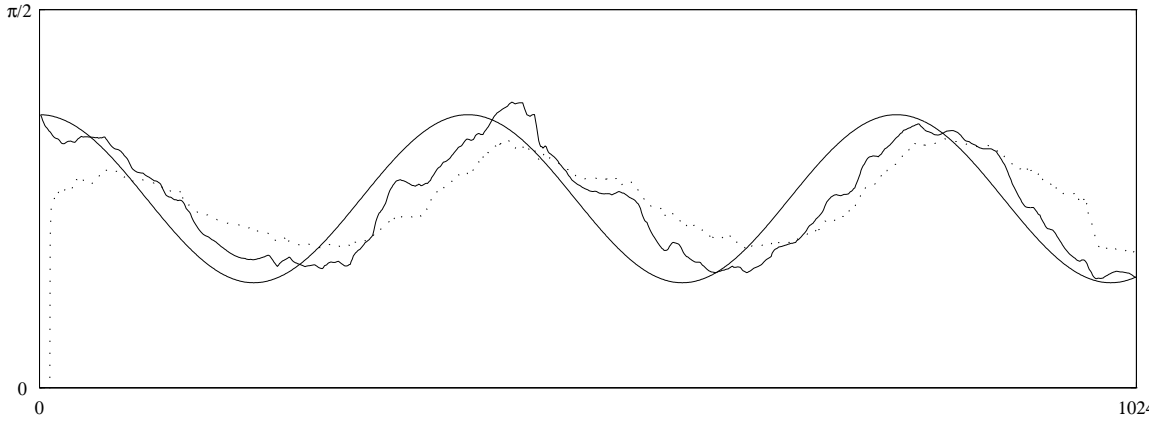

Figure 5.8: The angle of the true roots (solid smooth) with the estimated roots. Newton's method (solid) and the first order approximation method (dotted).

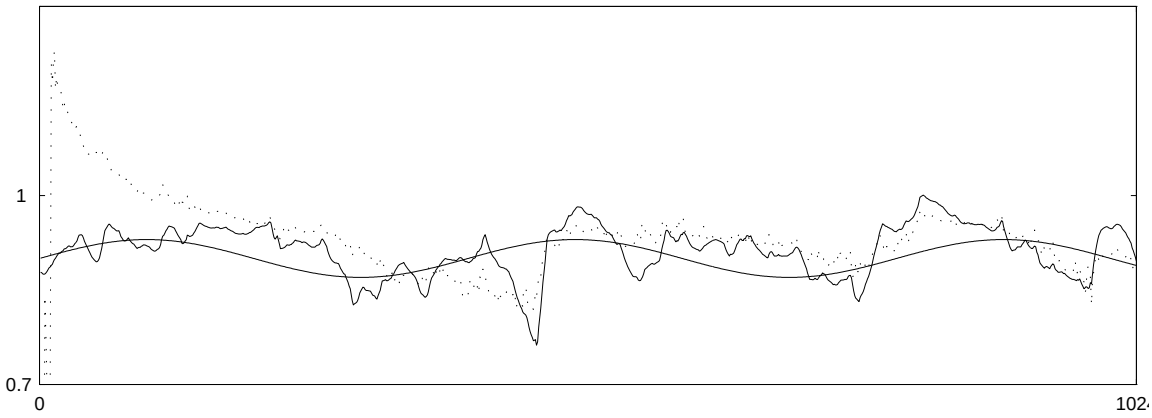

Figure 5.9: The moduli of the true roots (solid smooth) with the estimated roots. Newton's method (solid) and the first order approximation method (dotted).

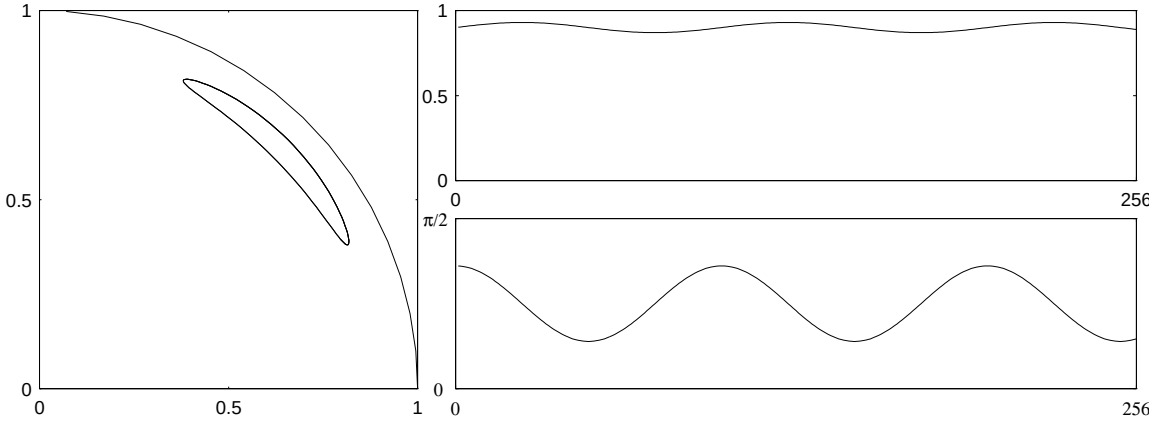

Figure 5.10: The root evolution in performance test of the Newton's method. Complex trajectory (left), magnitude (top) and angle (bottom).

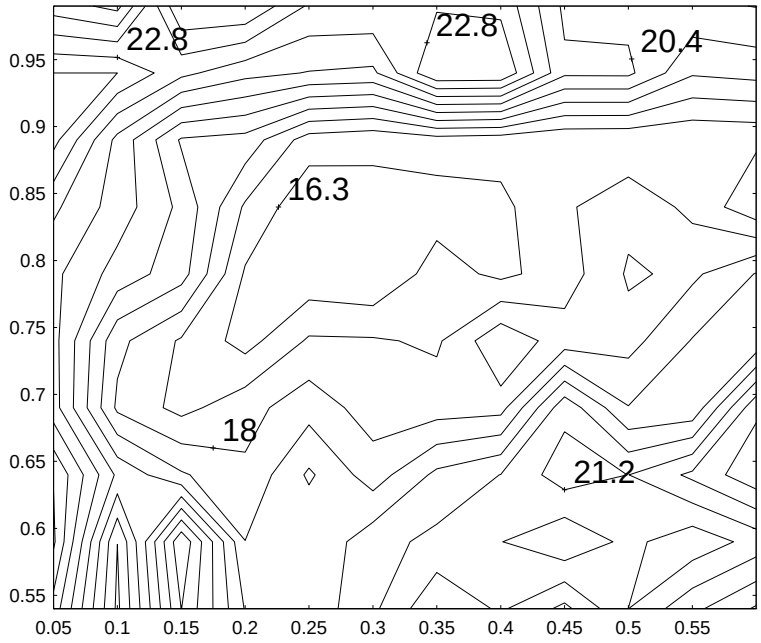

Figure 5.11: The contour of the error norm calculated with different α (horizontal) and λ (vertical) values.

CHAPTER VI

Application to EEG analysis

Autoregressive (AR) time series models have been successfully applied to the analysis of stationary EEG epochs since the end of the 1960's [10], [68], [14]. The ends to the modeling have been diverse, such as simulation [66], spectral analysis [14] and classification of sleep stages [16]. The usual way of dealing with the modeling of nonstationary EEG has been first to use either fixed or adaptive segmentation [4, 23]. The EEG is then treated as a stationary process within these segments. This approach has been proven to work satisfactorily in many cases, such as in sleep when the transistion periods of the EEG from one state to another are short compared with the stationary periods. Another such a case is when the EEG exhibits quite slow trends. An example of this is the slowly changing effects of medication.

Problems in nonstationary EEG analysis arise when the rate of change in EEG is too slow to be treated as abrupt but so fast that if we use segmentation the segments turn out to be so short that the estimates are useless due to their small sample properties. One obvious solution is to allow the AR coefficients to be time-varying.

There are two main classes of methods to solve the TVAR problem. The first is to use recursive estimation of the time-varying coefficient evolution and the second is to constrain the coefficient evolutions to be linear combinations of some basis functions with appropriate properties. These have been called the *stochastic* and *deterministic regression* approaches, respectively [15]. The stochastic approach is widely applied in biomedical signal analysis. These methods are also called adaptive methods. The most popular algorithms are the Least Mean Square algorithm, the Recursive Least Squares algorithm and the Kalman filter. These are introduced in Chapter 3. For use of Kalman filter in the analysis of EEG, see e.g. [5]. The main application of these methods has been the segmentation of the EEG. A review on the analysis of time-varying EEG with TVAR and other parametric models can be found in [28].

6.1 The ERS/ERD test

In many cases we are especially interested in tracking of narrow band characteristics of the EEG signal. One such a case is the event related synchronization/desynchronization (ERS/ERD) of alpha waves of EEG [44], [53]. When a patient has his/her eyes closed, the occipital EEG (primary visual cortex) shows high intensity in the 8–12 Hz region (alpha band, synchronization) while with the opening of the eyes this intensity decreases or even vanishes (desynchronization). In this kind of a situation we can assume that the EEG exhibits just one more or less rapid transition from a stationary state to another and that the transition starts at some short time after the visual stimulation. In the ERS/ERD test the end to the parametric modeling can be diverse, such as time-varying spectrum estimation or estimation of the delay or the rate of the transition.

There are three previous approaches to estimate the ERD/ERS evolution. The first is a short time Fourier transform -type (STFT) scheme in which discrete Fourier transforms are calculated from sliding window estimates of sample correlation [32]. In this method the window has to be made relatively short to maintain adequate time resolution and the variances of the time–frequency bins tend to become so large that the usefulness of the estimates suffers. We would then have to

average the individual estimates and the information on the differences between the estimates would be lost.

The second method is to pass the EEG through bandpass filters, square the outputs and average over realizations [53]. Thus we could obtain e.g. an estimate for the time–varying variance (power) of the alpha band. However, the bandpass filtering exhibits some drawbacks. The first is that the decay time of the filter impulse response (from 1 to 0.707) is approximately equal to the inverse of the width of the filter spectrum. If the band is 8–12 Hz, the decay time is about 0.25 s. In practice this means that the coefficient evolution estimates are effectively convolved with a window of this duration. The other problem in the filtering approach is that we are not able to track changes in the center frequency of a band within the passband. To overcome this problem we could divide the alpha band into sub–bands, but this would further decrease the time resolution.

The third method is based on deterministic Least Squares fitting of optimally selected basis functions to TVAR model [29].

In this chapter we propose the method with which the time-varying spectral characteristics of the ERD/ERS signal can be tracked. The method is based on using the time-varying ARMA model first to modeling the EEG signal. The roots of the parameters are then calculated with the root tracking algorithm based on the Newton's method. The calculated complex roots are further used for detection of the synchronized state.

6.2 Application

To test the proposed method we conducted a visual ERD/ERS test using the international 10-20 electrode system. This was carried out by opening and closing the eyes with an auditory stimulus (beep) on 15 second intervals. The subject was a healthy young female. Total amount of 20 transitions from desynchronized state to sunchronized state (eyes open $\rightarrow$ eyes closed) was measured. The channel of the best signal to noise ratio was then selected (the occipital O2 channel). The signal was low-pass filtered to allow for the decimation of the sampling frequency from 250 Hz to 250/4 Hz. The 20 measurements are shown in Fig. 6.1.

ARMA(p, q) model was then calculated with RLS using $p = 6$, $q = 2$ and $\lambda = 0.97$. The time-varying spectrum of the model is shown in figure Fig. 6.2. It is clearly seen that the model is able to estimate the alpha synchronization. The roots of the model denominator were then calculated with the Newton's method using the step size $\alpha = 0.1$. The complex roots of the first 3000 points are shown in Fig. 6.3. The roots are clearly clustered, and the discriminant (detection boundary) can be selected without difficulties. The imaginary part of all the roots are shown in Fig. 6.4. It can be notified that the detection could be based on the imaginary part only. The deep negative peaks in neighborhood of the time instant 5000 are due to complex root in second quadrant which is temporarily near the imaginary axis. The root has to be re-initialized if it tends to move out from the feasible region in the complex plane.

The result of the detection based on the boundary shown in Fig. 6.3 is shown in Fig. 6.5. The first 3000 points are shown. The detection indicator function is shown as staircase function together with the ongoing EEG signal. The detector output was filtered with symmetric median filter of length 51 points.

The result of the detection for the whole data set is shown in Fig. 6.6. The same data set was segmented with a segmentation program [19]. The result is also seen in Fig. 6.6. The means and the standard deviations for the time-delays measured from the trigger point were

$$t_{\mathrm{N}} = (59.6842 \pm 19.2361)\,\frac{\mathrm{s}}{64} \tag{6.1}$$

for the Newton's method and

$$t_{\mathrm{S}} = (55.3684 \pm 20.0955)\,\frac{\mathrm{s}}{64} \tag{6.2}$$

for the segmentation program. In seconds these are

$$t_{\mathrm{N}} = (0.9326 \pm 0.3006)\,\mathrm{s} \tag{6.3}$$

$$t_{\mathrm{S}} = (0.8651 \pm 0.3140)\,\mathrm{s} \tag{6.4}$$

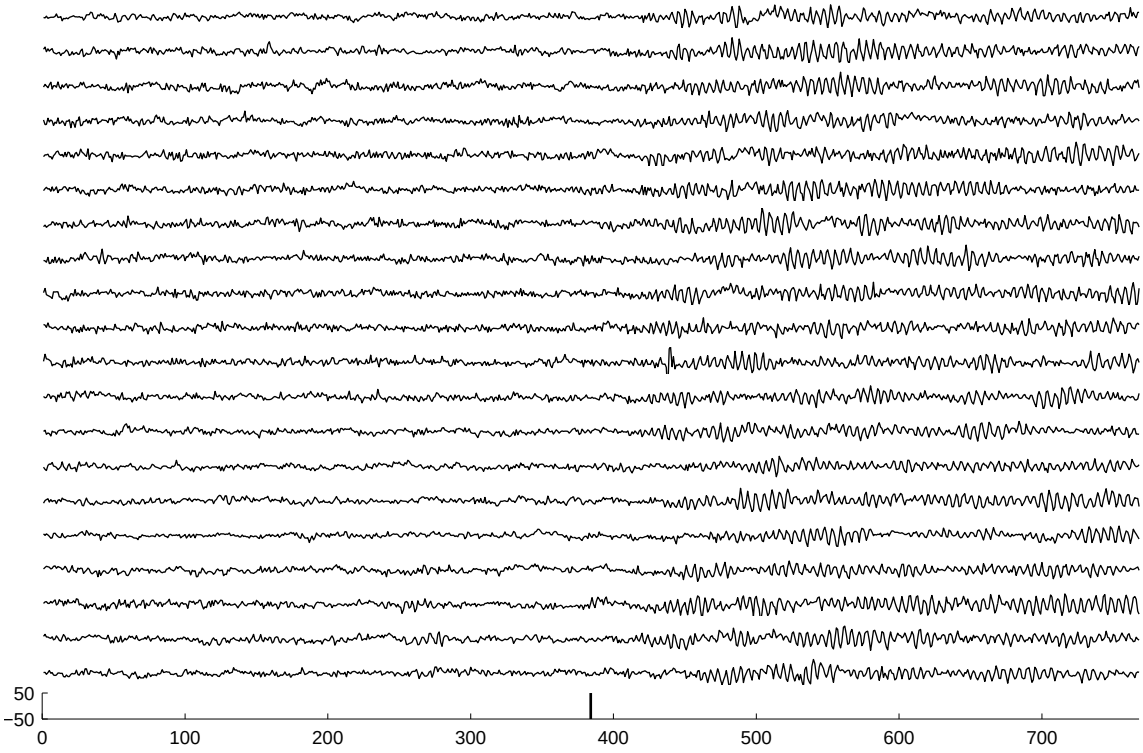

Figure 6.1: 20 ERS samples. The vertical line in scale axes is the auditory trigger (beep).

Samples 1, 2, 4, 14 and 16 were selected for closer investigation. They are shown in Fig. 6.7. The detection time points af the Newton's method and the segmentation program are shown with the detection made by visual inspection. As seen from samples there is some main differences in way of handling the data between different detection methods. The segmentation program compares the residuals of the models matched to different hypothesis about possible ghange times of the transition from one state to another. Linear models with piecewise constant parameters are used [20]. As seen in 4'th sample (3'rd in figure) the transition is not allways abrupt and the assumption of the piecewise constant linear model is not valid. In this sample the detection of the Newton's method is more accurate, because the detection is based on the spectral content of the signal. In 16'th sample (last in figure) the detection of the segmentation program is in coherence with the detection made by human. They seem to be reasonable when one investigates the morphology of the whole signal, like a human always do. However the signal seems to be morphologically more irregular between the time points 80 an 100 which means, that it may not contain very much information in alpha band. Merely it is due to increased noise power. In all the cases however differences between different detections are of the order of 20 points. This means only few periods in alpha activity. If any inference is to be made about the underlying neural processes, this kind of difference is not critical. In fact with narrow band processes the fluctuations of the instantaneous power of the signal can easily be of this length. This is clearly seen in first sample, which contains this kind of periods of deminished power.

The overall conclusion is that any detection method investigated in this study, including the one made by human, can not be said to be superior. This would necessitate the investigation of the properties of the ERD signal carefully. The investigation would give us the limits that could be acceptable in detection. After that the clinical relevancy of the test can be investigated.

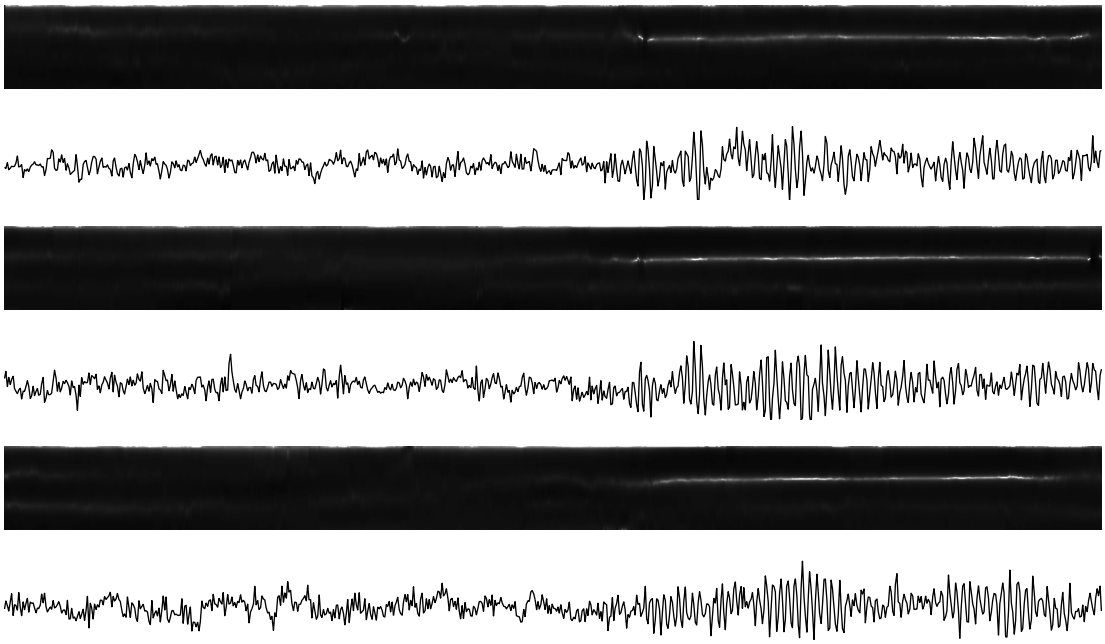

Figure 6.2: The time-varying spectrum of the three first ERS's.

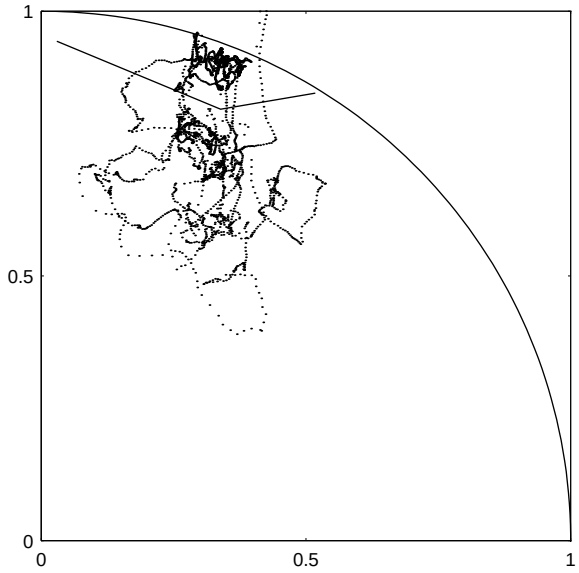

Figure 6.3: The first 3000 complex roots (dots) of the system calculated with Newton's method and the detection boundary (solid).

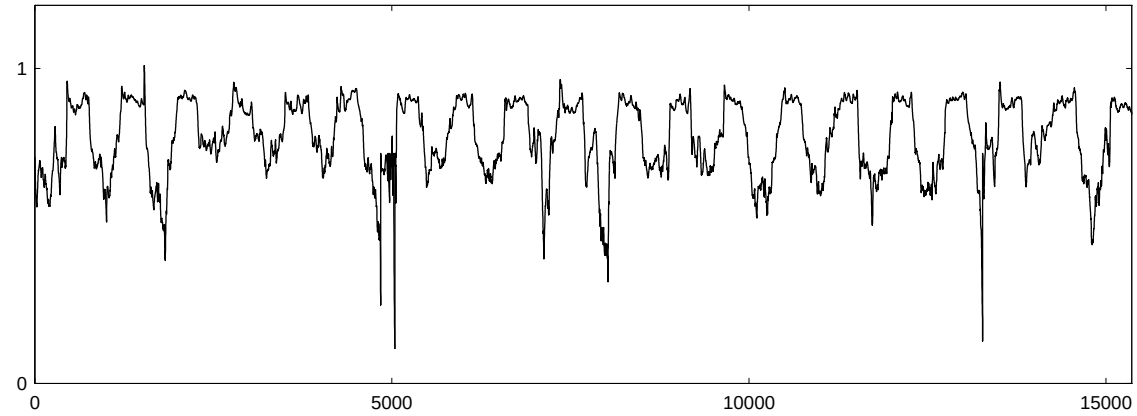

Figure 6.4: The imaginary part of all the roots.

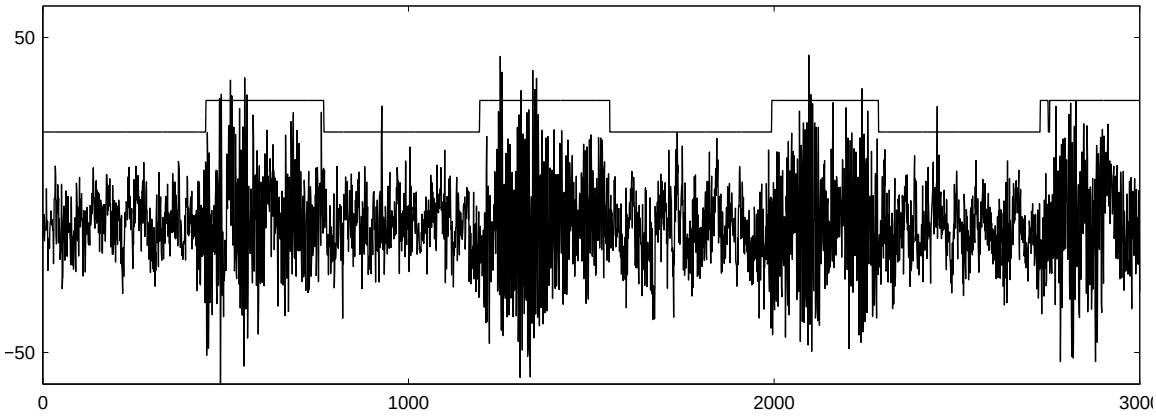

Figure 6.5: The EEG sample and the result of the detection (stair).
The first 3000 points are shown.

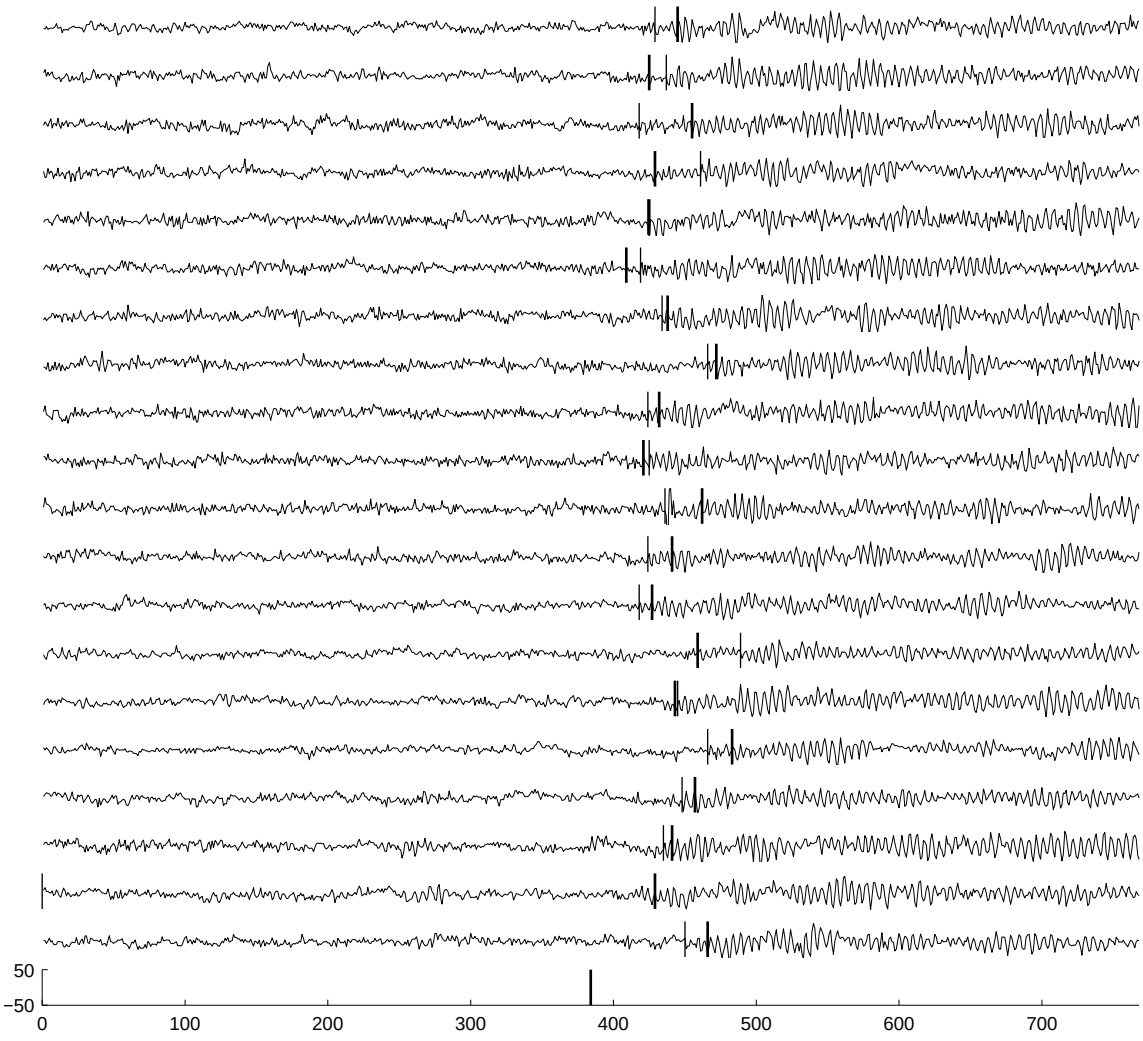

Figure 6.6: The result of the detection for the whole data set using the Newton's method (thick) and the estimated change point of the segmenter (thin).

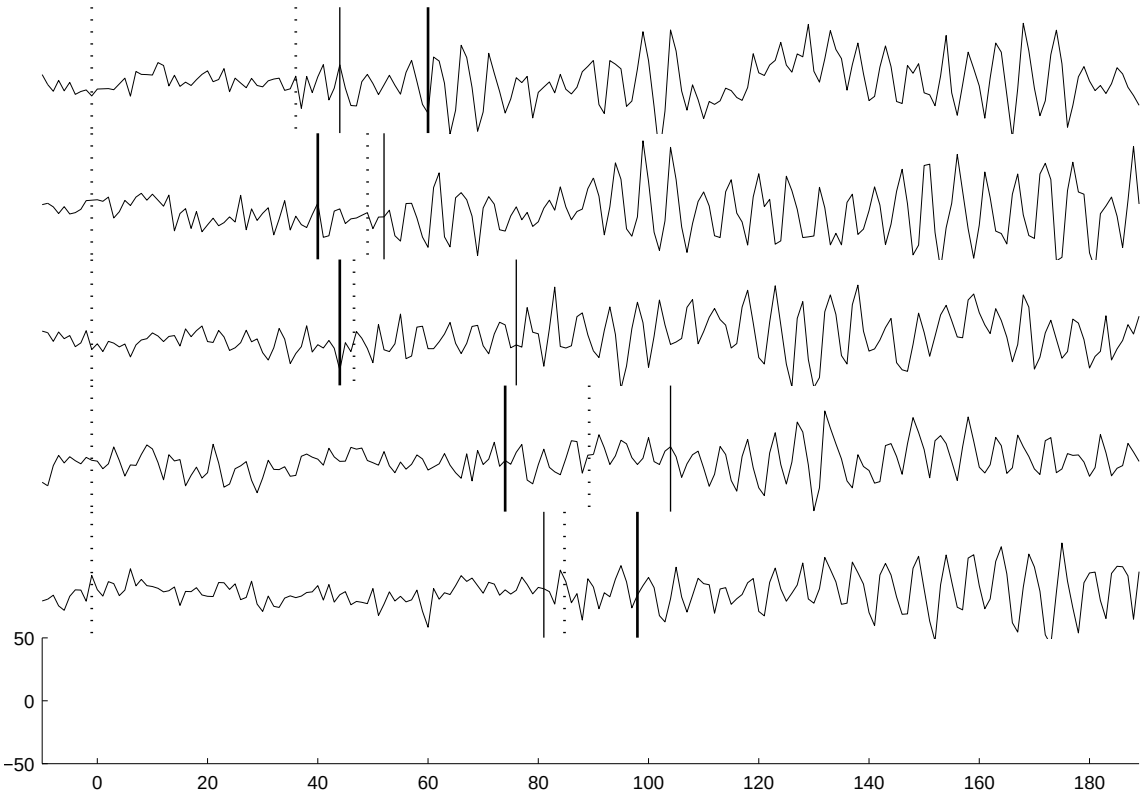

Figure 6.7: Five selected samples with the detected change points using the Newton's method (thick), the segmenter (thin) and the visual inspection (dotted,short). The stimulus is marked with the long dotted line.

CHAPTER VII

Summary and conclusions

In Chapter 2 we presented a review of several different approaches both to estimation of unknown parameters and to Bayesian estimation of random parameters. It was shown, that all the presented estimators can be seen as special cases of the weighted least squares problem. The solution of the least squares problem can be obtained directly in one linear operation in linear case and iteratively with the Gauss-Newton -method in nonlinear case. It was also seen that by taking into account the prior information about the parameters leads to Bayesian estimators. One such kind of an estimator is clearly the recursive mean square estimator of the state of a dynamical system. The solution of this is called the Kalman filter. As pointed out, the Kalman filter is then optimal in the same manner as the mean square estimator.

One of the results of Chapter 3 is the derivation of the different sequential estimation methods. The general algorithm for estimation of the parameters of the time-varying dynamical systems was presented first. Then the traditional recursive least squares algorithm was derived for the linear models based on the weighted least squares scheme. The optimal mean square solution for sequential estimation was then presented using Kalman filter. When the random walk model was used for the state-space -equations, the Kalman filter eqautions were seen to be identical to the most common recursive algorithms when the appropriate assumptions were made. We can thus conclude that the question of which method is the best one is usually irrelevant. The answer depends on the assumptions about the properties of the data and the properties of the underlying model for the parameters.

The time series modeling was introduced in Chapter 3 next. After a short review of stationary models, the time-varying case was introduced. The different time series model structures were connected to different adaptive algorithms with different choices of the regressor and parameter vectors. It should be noted that the optimality results derived in Chapter 2 are no longer valid. That is because the observation matrix is no longer deterministic, because it contains observations. We only conclude that the mean square estimator is not even unbiased unless the error v is uncorrelated with H [61]. This is seldom true with time-series models.

The survey of some possibilities for root tracking was presented in Chapter 4. Four methods were derived and presented in the same formalism as the general adaptive algorithms in Chapter 3. It was seen that the methods can be more or less systematic or be based on implementation of some traditional method for approximating the root of the polynomial. Several possibilities were not treated in this text. One of such a method is the use of the perturbation analysis of the eigenvalues of the companion matrix of the polynomial [30]. Many other methods for the solution or approximation of the matrix eigenvalues can be presented in sequential form. In the usual case the elements of the matrix would then be updated during the iteration.

In Chapter 5 some of the methods were evaluated using simulations. As pointed out, the simulation is not straightforward. The performance of a method is a function of all the properties of the underlying stochastic process, and thus the simulations has to be constrained to some class of the processes. We selected the AR(2)-processes having the root of the interest in the first quadrant of the complex plane. The fesible region depends clearly on the sampling rate of the signal and

the root of the interest can usually be moved to desirable region by interpolating and resampling the signal. Of greater importance is the rate of the movement of the root. When the simulation is supposed to give some idea of the performance of the algorithm in a real application, the rate has to be realistic for that situation. The simulations showed that the root tracking method which is based on the Newton's method is capable to track the movements of the complex roots when applied to time-varying polynomial coefficients. The Newton's method was also the most robust of all the methods evaluated.

In Chapter 6 the RLS-ARMA -modeling was used for the event related synchronization test. The end of the analysis was the detection of the starting time of the synchronization. We used the Newton's method for tracking the roots of the denominator polynomial of the ARMA(6,2)-model. Because the roots were clearly clustered in the complex plane, the detection was based on these roots. The results of the detection were found to be comparable to the results of a segmentation program and even to the results of a human expert. All of these were seen to have their own criterions for the detection, but none of them could be seen superior as compared to the others.

Generally we can conclude that the adaptive root tracking is one possibility for tracking the time-varying spectral properties of the signal, especially time-varying narrow band signals. The tracking capability of some methods is the same as the capability of the underlying adaptive algorithm, and thus the critical point is the selection of the time series model and the adaptive algorithm. Adaptive root tracking can be used for tracking the properties of the EEG-signal. This is the case at least with the ERS-test. However the inference about the underlying neural processes is not so straightforward. In some cases the dynamical (neural) system operates in such a mode, that the resulting activity is narrow band. In such cases, the state of the output activity of the dynamical system can fluctuate in time without the system state changes. In this kind of cases all the methods dealing with the power of the spectral components or the signal morphology, including the human investigation, can give uncorrect results. The possible solution for this is a global fitting of a narrow band time-varying model to signal [29].

APPENDIX I

Algorithms

Algorithm A.1: Root update with the first order Taylor approximation (4.27) used in zero tracking RLS (4.29) – (4.34).

```
function dz=da2dz(z,da)
% dz=da2dz(z,da);
% First order Taylor approximation of polynomial roots. If
% z=roots(a), dz \approx roots(a+da)-z. From Orfanidis et. al.
% "Zero-tracking adaptive filters", IEEE-ASSP, 34;1566-1572, 1986.

% P.A. Karjalainen, University of Kuopio, Mar 07. 1994

M=max(size(z));
z=z(:);
da=da(:);

for m=0:M
  nom=[nom z.^(M-m)];
end

zz=z*ones(1,M);

H=hankel([z;z]);
zzz=H(1:M,1:M);
P=zz-zzz;
P=P(:,2:M)';
P=[P;ones(1,M)];
PP=prod(P)';
den=PP*ones(1,M+1);
prt=-nom./den;
dz=prt*da;
```

Algorithm A.2: Zero tracking RLS algorithm (4.29) – (4.34) with first order Taylor approximation.

```
function [z,e,P,zz]=ztrls(x,n,w,initz,alpha,initP)
% [z,e,P,zz]=ztrls(x,n,w,initz,alpha,initP) ;
% Zero-tracking RLS. Uses the first order Taylor approximation of the
% roots of the polynomial. n is the order of the process. If the initial
% coefficient vector inita is not given inita=zeros(1:n+1). w is the
```

```
% root of exponential  window 0<w<=1.  Default  value of w is m=1, which
% is identical with  prewindowed LS estimate  of  the process. e is  the
% prediction  error  process.   aa  is  the  matrix  containing  all  AR
% coefficients during recursion.  P  is the last estimate of the inverse
% of the autocorrelation matrix. If the initial value initP is not given
% initP=inv(eps*eye(n)).

% P.A.Karjalainen, University of Kuopio, Finland, 26.05.1989

N=max(size(x));
if (nargin<4), z=initz(:);
 else a=zeros(1,n+1);a(1)=1;a(n+1)=0.5;z=roots(a); end
if nargin<3, w=1; end
if nargout==4, zz=zeros(N,n); end

x=x(:);
epsilon=eps;
I=eye(n);
if nargin<6,
 R0=epsilon*I;
 P=inv(R0);
else
 P=initP;
end

for i=(n+1):N
  aa=real(poly(z)).';
  a=aa(2:n+1);
  xx=x(i-1:-1:i-n);
  e(i)=x(i)+xx'*a;
  c=P*xx./(w+xx'*P*xx);            % Kalman gain vector c
  da=-e(i).*c;
  da=[0;da];
  dz=da2dz(z,da);
  z=z+alpha*dz;
  if nargout==4, zz(i,:)=z'; end
  P=(1/w)*(I-c*xx')*P;
end
```

Algorithm A.3: The classical Recursive Least Squares algorithm for time varying parameters of an ARMA(p, q) model (3.100) – (3.104).

```
function [hevol,e,hcov]=rlsarma(x,p,q,lambda,inith,initcov)
% [hevol,e,hcov]=rlsarma(x,p,q,lambda,inith,initcov);
% Calculates a recursive estimate hevol for time-varying  parameters  of
% an ARMA(p,q) model with  the classical RLS  algorithm.  lambda is the
% forgetting  factor (usually 0.9<lambda<1), inith  and  initcov are the
% initial estimates for the parameters and the corresponding covariance.
% If the  last two  are omitted, the data x is run backwards through the
% estimator to obtain valid initial estimates.

% 13.3.1995, J.P. Kaipio, Dept. of Applied Physics, U. of Kuopio
```

```
T=length(x);
hevol=zeros(p+q,T);
e=zeros(T,1);
if nargin<4, lambda=1; end
inilambda=0.97;
Tpre=6*1/(1-inilambda);
I=eye(p+q);

% Initialization: backward run
if nargin<6
  if nargin<5, inith=zeros(p+q,1); end
  hcov=eye(p+q);
  % Initialization of hevol always with inilambda so that
  % 3*1/(1-inilambda)=Tpre points will suffice
  hevol(:,min(Tpre,T-max(p,q)+1))=inith;
  for ii=min(Tpre,T-max(p,q)):-1:1
    xx=[x(ii+1:ii+p);e(ii+1:ii+q)];
    e(ii)=x(ii)-xx'*hevol(:,ii+1);
    K=hcov*xx./(lambda+xx'*hcov*xx);
    hevol(:,ii)=hevol(:,ii+1)+e(ii)*K;
    hcov=(I-K*xx')/lambda*hcov;
  end
else
  for ii=1:max(p,q), hevol(ii,:)=inith; end
  hcov=initcov;
end

% Main loop: forward run
for ii=max(p,q)+1:T
  xx=[x(ii-1:-1:ii-p);e(ii-1:-1:ii-q)];
  e(ii)=x(ii)-xx'*hevol(:,ii-1);
  K=hcov*xx./(lambda+xx'*hcov*xx);
  hevol(:,ii)=hevol(:,ii-1)+e(ii)*K;
  hcov=(I-K*xx')/lambda*hcov;
end

hevol=hevol';

%%  The evolutionary spectrum estimate of the model at time t
%%  is now
%%      H(omega;t)=freqz(\hat pe(t)*[1 h(t,p+1:p+q)],[1 -h(t,1:p)],128);
%%  where \hat pe(t) is a smoothed estimate of the prediction error standard
%%  deviation.
```

Algorithm A.4: Root tracking algorithm with direct root estimation (4.77) – (4.84).

```
function [th,e,P,Psi]=nehorai(y,Ncplx,Nreal,th0,P0,pl);
% [th,e,P,Psi]=nehorai(y,Ncplx,Nreal,th0,P0,pl);
% Direct recursive root tracking algorithm for time-varying AR
% coefficients of signal y using model with Ncplx complex pole pairs and
% Nreal real poles. th0 and P0 are the initial parameter vector and the
% covariancematrix, respectively.
```

```
% J.P. Kaipio, Univ. of Kuopio, Dept. of Applied Physics

if nargin<6,pl='y';end
% Total order and param. vector length:
N=2*Ncplx+Nreal;
y=y(:);
NL=length(y);
a=zeros(N+1,NL);
th=zeros(N,NL);
th(:,N)=th0(:);
% Initial values and coefficients:
a(:,N)=pr2c(th0,Ncplx,Nreal)';
if nargin<5, P=eye(N);else P=P0;end
psi=zeros(N,1);
Psi=zeros(N,NL);
phi=-y(1:N);
w=.97;
w0=.99;
winf=1;
%
e=zeros(NL,1);
L=zeros(N,1);
phi=-y(N:-1:1);
%
hold off
%figure(1),axis([1 NL 0 1.5])
%plot([1 NL],[th0(1) th0(1)],[1 NL],[th0(2) th0(2)]), hold on,drawnow
%
%figure(2),axis([1 NL -3 3]),hold on,drawnow
for t=N+1:NL
   e(t)=y(t)-a(2:N+1,t-1)'*phi;
   L=P*psi/(w+psi'*P*psi);
   P=(P-L*psi'*P)/w;
   th(:,t)=th(:,t-1)+L*e(t);
   %if t>255,e(t),L,L*e(t),P,a(:,t),agradth,pause,pause,end
   if pl=='y',
      clc,disp(['round ' num2str(t) 'of ' num2str(NL)])
   %   figure(1),plot([t-1 t],[th(1,t-1) th(1,t)]),drawnow
   %   disp([th(1,t-1) th(1,t)])
   %    P,L,psi,pause
   %   figure(2),plot([t-1 t],[e(t-1) e(t)]),drawnow
   end
   % Projection to th admissibility region:
     if Ncplx>0
       om=th(Ncplx+1:2*Ncplx,t);
       Iom=find(om<0);
       om(Iom)=-om(Iom);
       Iom=find(om>4/5*pi);
       om(Iom)=4/5*pi*ones(size(om(Iom)));
       th(Ncplx+1:2*Ncplx,t)=om;
       rho=th(1:Ncplx,t);
       Irho=find(rho<0);
       rho(Irho)=-rho(Irho);
       Irho=find(rho>.98);
```

```
        rho(Irho)=.98*ones(size(rho(Irho)));
        th(1:Ncplx,t)=rho;
      end
      if Nreal>0
        rl=th(2*Ncplx+1:N,t);
        Irl=find(rl<.1);
        rl(Irl)=0.1*ones(size(rl(Irl)));
        Irl=find(rl>.98);
        rl(Irl)=.98*ones(size(rl(Irl)));
        th(2*Ncplx+1:N,t)=rl;
      end
    % End of projection
    %
    a(:,t)=pr2c(th(:,t),Ncplx,Nreal)';
    for k=1:Ncplx
       agradth(:,k)=[-2*cos(om(k)); filter([-2*cos(om(k)) 2*rho(k)], ...
           [1 -2*rho(k)*cos(om(k)) rho(k)^2],a(2:N,t), ...
           [-4*rho(k)*(cos(om(k)))^2-2*cos(om(k))*a(2,t) 2*rho(k)])];
       agradth(:,Ncplx+k)=[2*rho(k)*sin(om(k)); ...
           filter(2*rho(k)*sin(om(k)), ...
           [1 -2*rho(k)*cos(om(k)) rho(k)^2],a(2:N,t), ...
           [4*rho(k)^2*cos(om(k))*sin(om(k))+2*rho(k)*sin(om(k))*a(2,t) 0])];
    end
    for k=1:Nreal
       agradth(:,2*Ncplx+k)=filter(-1,[1 -rl(k)],a(1:N,t));
    end
    phi=-y(t:-1:t-N+1);
    psi=agradth'*phi;
    Psi(:,t)=psi;
%    psi,pause
    % w=winf-(winf-w)*w0;
end
```

Algorithm A.5: Converts the polynomial roots to the coefficients.

```
function a=pr2c(th,Nc,Nr)
% a=pr2c(th,Nc,Nr);
% Polar form roots to coefficients mapping.

rho=th(1:Nc);
om=th(Nc+1:2*Nc);
rl=th(2*Nc+1:2*Nc+Nr);
a=[1];
for k=1:Nc
   z=[1 -rho(k)*exp(i*om(k))];
   a=conv(real(conv(z,conj(z))),a);
end
for k=1:Nr
   a=conv(a,[1 -rl(k)]);
end
```

Algorithm A.6: Root tracking with the Bairstow's method.

```
function rb=rbairstow(aa,Sinit,alpha)
% Polynomial root tracking using the Bairstow method
%
% Synopsis:
%  rb=rbairstow(aa,Sinit,alpha)
%
% Description:
% aa is given in predictor form. Sinit is a polynomial.
%
% Examples:
%
% See also:
% bairstow

%
% Pasi A. Karjalainen, Univ. of Kuopio, Dept. of Applied Physics
% Pasi.Karjalainen@uku.fi
%

[N,p]=size(aa);
S = Sinit;

for ii=1:N,
 rb(ii,:)=bairstow([1 -aa(ii,:)],S,1,alpha).';
 S=real(poly(rb(ii,:)));
end
```

Algorithm A.7: Bairtows method for finding the roots of the real polynomial introduced in Section 4.4.

```
function [rb,w,Jcond]=bairstow(A,Sinit,k,alpha)
% Polynomial root finding using the Bairstow method
%
% Synopsis:
%  [rb,w,Jcond]=bairstow(A,Sinit,k,alpha)
%
% Description:
% Function calculates k steps of Bairstow algorithm starting
% from roots of second order polynomial Sinit.
% Can be used for tracking.
%
% Examples:
% A=mkpol([0.8 0.2*pi;0.8 0.6*pi]);
% S1=mkpol([0.7 0.1*pi]);
%       S = S1;
% for i=1:10,
%  rb = bairstow(A,S,1,0.5);
%  plot(real(roots(A)),imag(roots(A)),'y+',...
%       real(roots(S1)),imag(roots(S1)),'r+',...
%       real(rb),imag(rb),'g.')
%  setunit,drawnow
%  S = poly(rb);
% end
%
```

```
% See also:
% setunit

%
% Pasi A. Karjalainen, Univ. of Kuopio, Dept. of Applied Physics
% Pasi.Karjalainen@uku.fi
%

if nargin<4, alpha=1; end
p=length(A);
u = -Sinit(2);
v = -Sinit(3);
w = [ u ; v ];
S = Sinit;

for i=1:k,
 [B,R]=deconv(A,S);
 r = A(p-1) + u*B(p-2) + v*B(p-3);
 s = A(p) + r*u + v*B(p-2);
 f=[ r ; s];
 J=[B(p-2) B(p-3) ; B(p-2)*u+r B(p-3)*u+B(p-2)];
 % J=[B(p-2) B(p-3) ; r B(p-2)];
 if nargout>2, Jcond=cond(J); end
 w(:,i+1) = w(:,i) - alpha*inv(J)*f;
 u = w(1,i+1);
 v = w(2,i+1);
 S = [1 -u -v];
end
rb = roots(S);
```

Algorithm A.8: Root tracking with the Newton's method (4.12).

```
function r=rnewton(A,r0,alpha,imin)
% r=rnewton(A,R0,alpha);
% traces a root of the polynomials  contained  in the rows of A starting
% from  R0. The root is approximated using the Newton (1st) order method
% for  fixed point iteration  of a  (complex) zero  of  a  function. The
% method  is applicable if the coefficients of successive polynomials do
% not change much  from row  to  row, i.e. the column  vectors of  A are
% slowly varying alpha is the step  size, if omitted, it is set to unity
% giving the standard Newton method.

% Copyright 28.12.1992  J. Kaipio

[N,p]=size(A);
if nargin<3,alpha=1;end

f=polyval([1 -A(1,:)],r0);
fp=polyval([p:-1:1].*[1 -A(1,1:p-1)],r0);
if abs(fp)>1e-2,r(1)=r0-alpha*f/fp;else r(1)=r0;end

for i=1:N-1
  f=polyval([1 -A(i+1,:)],r(i));
  fp=polyval([p:-1:1].*[1 -A(i+1,1:p-1)],r(i));
```

```
  if abs(fp)>1e-2,r(i+1)=r(i)-alpha*f/fp;else r(i+1)=r(i);end
%  if (imag(r(i+1)) < imin),r(i+1)=real(r(i+1))+imin*sqrt(-1);end
%  if (real(r(i+1)) < imin),r(i+1)=imin + imag(r(i+1)) ;end
  if (imag(r(i+1)) < imin),r(i+1)=r0;end
  if (real(r(i+1)) < 0),r(i+1)=r0;end
end
I=find(imag(r)<0);r(I)=conj(r(I));
r=r(:);
```

[1] S.T. Alexander and V. Stonick. Fast adaptive polynomial root tracking using a homotopy. In *IEEE Int. Conf. Acoust. Speech Signal Processing*, pages III–480 – III–483, 1993.

[2] T.M. Apostol. *Mathematical Analysis*. Addison-Wesley, 1974.

[3] Å. Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[4] G. Bodenstein and H.M. Praetorius. Feature extraction from the electroencephalogram by adaptive segmentation. *Proc IEEE*, 65:642–652, 1977.

[5] T. Bohlin. Analysis of EEG signals with changing spectra using a short-word Kalman estimator. *Math Biosci*, 35:221–259, 1977.

[6] B.P. Carlin and T.A. Louis. *Bayes and empirical Bayes methods for data analysis*. Chapman & Hall, 1996.

[7] D.G. Childers, editor. *Modern Spectrum Analysis*. IEEE Press, 1978.

[8] B. Choi. *ARMA Model Identification*. Springer-Verlag, 1992.

[9] H. Cramer. *Mathematical Methods in Statistics*. Princeton University Press, 1946.

[10] P.B.C. Fenwick, P. Mitchie, J. Dollimore, and G.W. Fenton. Application of the autoregressive model to E.E.G. analysis. *Agressologie*, 10:553–564, 1969.

[11] A.B. Fineberg and R.J. Mammone. A method for instantaneous frequency tracking of multiple narrowband signals. *Signal Processing*, 29:29–44, 1992.

[12] B. Friedlander. Lattice filters for adaptive processing. *Proc IEEE*, 70:829–867, 1982.

[13] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian data analysis*. Chapman & Hall, 1995.

[14] W. Gersch. Spectral analysis of EEG's by autoregressive spectral decomposition of time series. *Math Biosci*, 7:205–222, 1970.

[15] W. Gersch. Non-stationary multichannel time series analysis. In *Methods of Analysis of Brain Electrical and Magnetic Signals*, volume 1 of *Handbook of Electroencephalography and Clinical Neurophysiology*, pages 261–296. Elsevier, 1987.

[16] W. Gersch, J. Yonemoto, and P. Naitoh. Automatic classification of multivariate EEGs using an amount of information measure and the eigenvalues of parametric time series model features. *Comput Biomed Res*, 10:297–318, 1977.

[17] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1989.

[18] L. Guo, L. Ljung, and P. Priouret. Tracking performance analysis of the forgetting factor RLS algorithm. Technical Report LiTH-ISY-I-1393, Linköping University, 1992.

[19] F. Gustafsson. A change detection and segmentation toolbox for Matlab. Technical Report LiTH-ISY-I-1669, Linköping University, 1994.

[20] F. Gustafsson. Segmentation of signals using piecewise constant linear regression models. Technical Report LiTH-ISY-R-1672, University of Linköping, Department of Automatic Control, 1994.

[21] F. Gustafsson and S. Gunnarsson L. Ljung. On time-frequency resolution of signal properties using parametric techniques. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 2259–2264, Orlando, Florida, 1994.

[22] P.S. Hamilton and W.J. Tompkins. Detection of ventricular fibrillation and tachycardia by adaptive modelling. In *IEEE EMBC-87*, pages 1881–1882, 1987.

[23] A. Hasman, B. Jansen, G. Landeweerd, and A. van Blokland-Vogelesang. Demonstration of segmentation techniques for EEG records. *Int J Bio-Med Comput*, 9:311–321, 1978.

[24] S. Haykin. *Adaptive filter theory*. Prentice Hall, 2 edition, 1991.

[25] N.J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 1996.

[26] M.L. Honig and D.G. Messerschmidt. *Adaptive Filters: Structures, Algorithms and Applications*. Kluwer Academic Publishers, 1984.

[27] J.E. Dennis Jr. and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, 1983.

[28] J.P. Kaipio. *Simulation and Estimation of Nonstationary EEG*. PhD thesis, University of Kuopio, 1996.

[29] J.P. Kaipio and P.A. Karjalainen. TVAR modeling of event-related synchronization changes. The optimal basis approach. Technical report, Department of Applied Physics, University of Kuopio, 1995. To appear also in *IEEE Trans Biomed Eng*.

[30] J.P. Kaipio, P.A. Karjalainen, and M. Juntunen. Perturbation expansions in polynomial root tracking. Technical Report 2/96, University of Kuopio, Department of Applied Physics Report Series, 1996.

[31] A. Kangas. Algorithms for adaptive factorization of polynomials. *Signal Processing*, 35:67–74, 1994.

[32] N. Kawabata. A nonstationary analysis of the electroencephalogram. *IEEE Trans Biomed Eng*, 20:444–452, 1973.

[33] J.P. Keating, R.L. Mason, and P.K. Sen. *Pitman's measure of closeness: a comparison of statistical estimators*. SIAM, 1993.

[34] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.

[35] S.B. Kesler, editor. *Modern Spectrum Analysis, II*. IEEE Press, 1986.

[36] G. Kitagawa and W. Gersch. A smoothness priors time-varying AR coefficient modeling of nonstionary covariance time series. *IEEE Trans Automat Contr*, 30:48–56, 1985.

[37] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[38] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

[39] L. Ljung and S. Gunnarsson. Adaptation and tracking in system identification – a survey. *Automatica*, 26:7–21, 1990.

[40] L. Ljung, G. Pflug, and H. Walk. *Stochastic Approximation and Optimization of Random Systems*. Birkhäuser, 1992.

[41] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, 1983.

[42] L.T. Mainardi, A.M. Bianchi, G. Baselli, and S. Cerutti. Pole-tracking algorithms for the extraction of time-variant heart rate variability spectral parameters. *IEEE Trans Biomed Eng*, 42(3):250–259, march 1995.

[43] J. Makhoul. Stable and efficient lattice methods for linear prediction. *IEEE Trans Acoust, Speech Signal Processing*, 25:423–428, 1997.

[44] O.M. Markand. Alpha rhythms. *J Clin Neurophysiol*, 7:163–189, 1990.

[45] S.L. Marple. *Digital Spectral Analysis*. Prentice-Hall International, 1987.

[46] J.L. Melsa and D.L. Cohn. *Decision and Estimation Theory*. McGraw-Hill, 1978.

[47] R.E. Mortensen. *Random Signals and Systems*. Wiley, 1987.

[48] A. Nehorai and D. Starer. Adaptive pole estimation. *IEEE Trans Acoust, Speech Signal Processing*, 38:825–838, 1990.

[49] S. Orfanidis and L. Vail. Zero-tracking adaptive filters. *IEEE Trans Acoust, Speech Signal Processing*, 34:1566–1572, 1986.

[50] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1984.

[51] L. Patomäki, J.P. Kaipio, and P.A. Karjalainen. Tracking of nonstationary EEG with the roots of ARMA models. In *IEEE EMBS'95*, 1995.

[52] L Patomäki, J.P. Kaipio, P.A. Karjalainen, and M. Juntunen. Tracking of nonstationary EEG with the polynomial root perturbation. In *EMBS'96*, 1996.

[53] G. Pfurtscheller, C. Neuper, and W. Mohl. Event-related desynchronization (ERD) during visual processing. *Int J Psychophysiol*, 16:147–153, 1994.

[54] M.B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1981.

[55] J.A. Rice. *Mathematical statistics and data analysis*. Duxbury Press, 2 edition, 1995.

[56] Z. Rogowski, I. Gath, and E. Bental. On the prediction of epileptic seizures. *Biol Cybern*, 42:9–15, 1981.

[57] A.H. Sayed and T. Kailath. A state-space approach to adaptive RLS filtering. *IEEE Signal Processing Magazine*, 11:18–60, 1994.

[58] T. Shan and T. Kailath. Directional signal separation by adaptive arrays with a root-tracking algorithm. In *IEEE Int. Conf. Acoust. Speech Signal Processing*, pages 2288–2291, 1987.

[59] L.H. Sibul, editor. *Adaptive signal processing*. 1987.

[60] V. Solo and X. Kong. *Adaptive Signal Processing Algorithms. Stability and Performance*. Wiley, 1995.

[61] H.W. Sorenson. *Parameter Estimation: Principles and Problems*. Marcel Dekker, 1980.

[62] H.W. Sorenson, editor. *Kalman Filtering: Theory and Applications*. IEEE Press, 1985.

[63] D. Starer and A. Nehorai. Path-following algorithm for passive localization of near-field sources. In *ASSP Workshop on Spectrum Estimation and Modelling*, pages 322–326, 1990.

[64] D. Starer and A. Nehorai. Adaptive polynomial factorization by coefficient matching. *IEEE Trans Signal Processing*, 39:527–530, 1991.

[65] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, 1980.

[66] A. Wennberg and A. Isaksson. Simulation of nonstationary EEG signals as a means of objective clinical interpretation of EEG. In *Quantitative Analytical Studies in Epilepsy*, pages 493–509. Raven Press, 1976.

[67] B. Widrow and S.D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, 1985.

[68] L. Zetterberg. Estimation of parameters for a linear difference equation with application to EEG analysis. *Math Biosci*, 5:227–275, 1969.