VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNíHO INŽENÝRSTVí
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

THE USE OF RECURSIVE LEAST SQUARES METHOD FOR VEHICLE DYNAMICS ANALYSIS

VYUŽITÍ REKURZIVNÍ METODY NEJMENŠÍCH ČTVERCŮ PRO ANALÝZU DYNAMIKY VOZIDEL

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE AUTHOR	Bc. PAVLA SLADKÁ
VEDOUCÍ PRÁCE SUPERVISOR	Ing. PETR PORTEŠ, Dr.

BRNO 2010

Licenční smlouva
poskytovaná k výkonu práva užít školní dílo
uzavřená mezi smluvními stranami:

1. Paní

Jméno a příjmení:	Bc. Pavla Sladká
Bytem:	Nádražní 1260, 664 34, Kuřim
Narozena (datum a místo):	3. 12. 1985, Brno

(dále jen autor)

a

2. Vysoké učení technické v Brně

Fakulta strojního inženýrství
se sídlem Technická 2896/2, 61669, Brno - Královo Pole
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
. . .

(dále jen nabyvatel)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   - ☐ disertační práce
   - ☒ diplomová práce
   - ☐ bakalářská práce
   - ☐ jiná práce, jejíž druh je specifikován jako ....

   (dále jen VŠKP nebo dílo)

Název VŠKP:	The Use of Recursive Least Squares Method for Vehicle Dynamics Analysis
Vedoucí/ školitel VŠKP:	Ing. Petr Porteš, Dr.
Ústav:	Ústav matematiky
Datum obhajoby VŠKP:	23. 6. 2010

   VŠKP odevzdal autor nabyvateli v[1]:

   - ☐ tištěné formě — počet exemplářů 2
   - ☐ elektronické formě — počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

[1]hodící se zaškrtněte

Čl. 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti

   □ ihned po uzavření této smlouvy

   □ 1 rok po uzavření této smlouvy

   □ 3 roky po uzavření této smlouvy

   □ 5 let po uzavření této smlouvy

   □ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením §47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Čl. 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

Nabyvatel Autor

Summary

This diploma thesis amplifies the theoretical bases required to design the recursive least squares algorithm and, in consequence, its application to the experimental data measured during test manoeuvre realized in 2001. A lateral dynamics of single-track planar model of vehicle was analyzed. It contains also a comparing of the results obtained by the recursive algorithm and Kalman filter algorithm.

Abstrakt

Tato diplomová práce nastiňuje teoretické základy potřebné pro návrh algoritmu rekurzivní metody nejmenších čtverců a následně jeho aplikaci na experimentální data naměřená při testovacím manévru uskutečněném v roce 2001. Analyzována byla příčná dynamika jednostopého rovinného modelu vozidla. Práce také obsahuje srovnání výsledků získaných jednak rekurzivním algoritmem a dále i algoritmem Kalmanova filtru.

Keywords
dynamic system, white noise, least squares method, single-track model, adaptive filter, recursive algorithm, filter coefficients

Klíčová slova
dynamický systém, bílý šum, metoda nejmenších čtverců, jednostopý model, adaptivní filtr, rekurzivní algoritmus, koeficienty filtru

SLADKÁ, P. *The Use of Recursive Least Squares Method for Vehicle Dynamics Analysis.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 69 s. Vedoucí diplomové práce Ing. Petr Porteš, Dr.

Prohlašuji, že jsem diplomovou práci *The Use of Recursive Least Squares Method for Vehicle Dynamics Analysis* vypracovala samostatně pod vedením Ing. Petra Porteše, Dr. s použitím materiálů uvedených v seznamu literatury.

Bc. Pavla Sladká

I would like to thank my supervisor Ing. Petr Porteš, Dr. for his advice and help during writing this thesis. Hearty thanks for support during all my studies on the university belong to my family and close friends.

Bc. Pavla Sladká

Table of contents

1 Introduction **3**
1.1 Motivation . . . 4

2 Essential Knowledge for System Identification **7**
2.1 Dynamic Systems . . . 7
2.2 Term of A Model . . . 7
2.3 The System Identification Procedure . . . 8
2.3.1 Three basic items . . . 8
2.3.2 Validation of the model . . . 9
2.3.3 The System Identification Loop . . . 9

3 Linear Dynamic Systems **11**
3.1 Continuous-time Models . . . 12
3.2 Discrete-time Models . . . 14

4 Stochastic Systems **17**
4.1 Short View into Probability Theory . . . 17
4.1.1 Characteristics of Random Variable . . . 17
4.1.2 Parameter Estimation: Bias . . . 18
4.1.3 Stochastic (Random) Processes and Their Characteristics . . . 19
4.2 Linear Stochastic Systems . . . 20
4.2.1 Continuous-time Models . . . 21
4.2.2 Discrete-time Models . . . 22

5 Parameter Estimation Methods **25**
5.1 Linear Least Square Method (LLS) . . . 25
5.2 Nonlinear Least Squares Method (NLS) . . . 27
5.3 Weighted Least Squares Method (WLS) . . . 29

6 Adaptive Filtering **31**
6.1 The Recursive Least Squares Method (RLS) . . . 32
6.1.1 Recursive Algorithm . . . 33
6.1.2 Initialization of RLS Algorithm . . . 36
6.1.3 Choice of the Forgetting Factor . . . 37

7 Modeling of Vehicle Dynamics **39**
7.1 Tire Model 39
7.2 Single-Track Model 40

8 Experiment **45**
8.1 Test Car 45
8.2 Test Track 46
8.3 Measuring Equipment 46

9 Application of Recursive Algorithm **49**
9.1 Model 49
9.2 MATLAB Files 50
9.2.1 M-file for Recursive Algorithm 51
9.2.2 M-file for Discretization of State Equation 51
9.3 Results 54
9.3.1 Results for State Variables 54
9.3.2 Derived Vehicle Position from Estimated State Variables 57
9.4 Comparison with Kalman Filter Results 57

10 Conclusion **61**

A Evolutions of Derived Measured Signals **63**

Chapter 1

Introduction

The goal of this thesis is to outline the theoretical basics needful for understanding the principle of adaptive filter behaviour and consequently its application to the real problem. More precise, we focus on the adaptive filter relied on the recursive least squares algorithm and its usage to analysis of vehicle driving conditions. For this purpose we design the flow diagram for processing of data obtained from the experimental measuring.

The thesis is segmented into eight chapters except the introduction including a short motivation, conclusion, and appendix. Chapter 2 deals with the system identification, explains dynamic systems, the term of model with its classification, and the identification procedure.

It is necessary to realize that almost every physical process is in fact a dynamic system. We must know the mathematical description of the dynamic system to express the evolution of its characteristic in time. Chapter 3 is thus devoted to the mathematical models both continuous-time and discrete-time of linear dynamic system description. The following chapter analyze these models with added random components, i.e. deals with the linear stochastic systems. Some required notes from probability theory are stated in this part, especially characteristics of random variables and processes, and discussion about the evolution of the expected value and covariance of the state and output vector.

But, in dynamic system, we are not able to measure every time all magnitudes which we want to control. Hence, we have to estimate the state and output of the system which are function of the measuring. The estimation methods are specified in the chapter 5. The first mentioned method is the linear least squares method, subsequently the nonlinear and weighted least squares method. The least squares method was the first method for formulation of the optimal estimate from noisy data. Carl Friedrich Gauss is credited with developing the fundamentals of the basis for least-squares analysis in 1795 at the age of eighteen. Gauss did not publish the method until 1809 and the idea of least-squares analysis was also independently formulated by the Frenchman Adrien-Marie Legendre in 1805, who was the first to publish the method, and the American Robert Adrain in 1808.

Chapter 6 gives attention to the adaptive filter which relies on the recursive least squares algorithm. It contains the derivation of recursive algorithm, its initialization and also the choice of forgetting factor. This algorithm is used to find the filter coefficients that relate to recursively producing the least squares of the error signal (difference between the desired and the actual signal). This is contrast to other algorithms that aim to reduce the mean squared error. The difference is that recursive least squares filters are dependent on the signals themselves, whereas mean squared error filters are dependent on their statistics (specifically, the autocorrelation of the input and the cross-correlation of the input and desired signals). If these statistics are known, a mean squared error filter with fixed coefficients, i.e. independent of the incoming data, can be built.

Chapter with the title Modeling of Vehicle Dynamics is divided into two parts. The first part deals with the tire model and explains which forces are generated as the tire rolls. The second one defines the term of the single-track model, also known as a bicycle model, which serves to investigate theoretically the lateral dynamics of the vehicle in the horizontal plane. This model will be later used in MATLAB simulation of experimental data.

The following chapter describes the test car, defines the test track, and shows a measuring equipment utilize during the test manoeuver. This experiment was realized in 2001 by the Institute of Forensic Engineering of Brno University of Technology in conjunction with the Department of Transporting Technology of Brno University of Technology. This rubric also includes three graphs illustrating the evolution of the measured signals. The derived signals are stated in the appendix at the end of this thesis.

Application of the Recursive Algorithm is the title of the last chapter, which consists of four basic sections. The beginning of the chapter treats primarily of the discretization of the single-track model and discuss a sampling period of measured time--variant variables that occurred in this bicycle model. Next section covers m-files with the recursive algorithm, and discretization of the single-track model. Also it contains illustrating figures of the output of created MATLAB program. The most important part of this chapter and maybe of all this thesis is given as penultimate. This section summarize the results obtained by MATLAB simulation and compare the evolutions of the filtered out (estimated) and measured outputs. The last part is a comparing of results acquired by two different algorithms, namely by our recursive algorithm and Kalman filter algorithm.

1.1 Motivation

Suppose that a signal $y(t)$[1] is received with transmitter noise, more precisely with **white noise**. We will attempt to recover the desired signal $y(t)$ by using an adaptive filter (chapter 7), $\boldsymbol{\theta}$

$$\hat{y}(t|\boldsymbol{\theta}) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} = \boldsymbol{\theta}^T\boldsymbol{\varphi}(t). \tag{1.1}$$

This expression is predicted output at time t, where the vector of regressors $\boldsymbol{\varphi}(t)$ has the form[2]

$$\boldsymbol{\varphi}(t) = [u_1(t) \quad \dots \quad u_m(t) \quad y(t-1) \quad \dots \quad y(t-n)]$$

[1]We shall generally denote the inputs and outputs of the systems at time t by $u(t)$ and $y(t)$, respectively.

[2]This form can be expressed as $\boldsymbol{\varphi}(t) = [\boldsymbol{u}^T(t) \quad y(t-1) \quad \dots \quad y(t-n)]$ with the $m \times 1$ input vector $\boldsymbol{u}^T(t) = [u_1(t) \quad \dots \quad u_m(t)]$.

and the unknown parameter $\boldsymbol{\theta}$ is following

$$\boldsymbol{\theta} = [a_1 \quad \ldots \quad a_m \quad b_1 \quad \ldots \quad b_n].$$

In other words, the equation (1.1) expresses, that the predictor $\hat{y}(t|\boldsymbol{\theta})$ at time t is a linear combination of the inputs at time t, the outputs of previous iterations, and filter coefficients in vector $\boldsymbol{\theta}$.

Our goal is to estimate the filter coefficients, i.e. vector $\boldsymbol{\theta}$, with the knowledge about the input-output data and by using the least squares method (chapter 5).

When the parameter of the filter, $\boldsymbol{\theta}$, is estimated, at each time t we refer to the new least squares estimate $\boldsymbol{\theta}_t$. As time evolves, we would like to avoid completely redoing the least squares algorithm to find the new estimate for $\boldsymbol{\theta}_{t+1}$ in terms of $\boldsymbol{\theta}_t$. For this purpose we shall introduce the recursive least squares algorithm (chapter 6) and by using this we shall analyze the vehicle dynamics.

Remark. We can call the model (1.1) as an archetypical problem. In our case the meaning of the word archetypical is that we described this original model like some prototype and this model will be used as representative over all this thesis. But, of course, we have to notify that this prototype can be modified for certain requirements.

Chapter 2

Essential Knowledge for System Identification

System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the system. The subject is thus part of basic scientific methodology, therefore dynamical systems are abundant in our environment. The techniques of system identification have a wide area of application. At the creation of this chapter we gathered especially from [8] and [13].

2.1 Dynamic Systems

In loose terms a **system** is an object in which variables of different kinds interact and produce observable signals. The observable signals in which we are interested are usually called **outputs** and are measurable. The system is also affected and influenced by external stimuli. External signals that can be manipulated by the observer are called **inputs**, which are known or at least measurable and controllable. Others are called **disturbances** and can be divided into those that are directly measured and those that are only examined through their effect on the output. The distinction between inputs and measured disturbances is very often less important for process of modeling.

Dynamic system consists of the **state space** and **dynamic conditions**. The coordinates of the state space give account of the system at a given time and dynamic conditions describe the change of the system. Then system state is described by the **state vector** that all lies in the state space. The dynamic conditions are often given by the system of differential or difference equations which give account of the change of the state vector at given time. Mathematical models are instrumental to obtain the solution that express the functional relation between the state variables and the initial conditions coupled with inputs.

2.2 Term of A Model

We can call the subject of research (the system) the **original** and his representation (which is also some system) is called a **model**. Each model is created for some purpose.

During the process of modeling we preserve only the components, properties, relationships of the system which are substantial for this purpose and we do not deal with those which do not play the important role in the system.

Classification of Mathematical Models

The basic classification of mathematical (analytical, computer) models is based on their behaviour at time. In light of this we recognize these two types of models

- **static** (**time-invariant**) - these models do not have changes in time,
- **dynamic (time-variant)** - this sort of models develops in time.

Dynamic models we can be further distinguished into following types of models

- ⋆ **without memory** - in this case the state at posterior instant of time depends only on the current state (so called Markov models),
- ⋆ **with memory** - moreover, compared to the models without memory, depends on the state at some past instant of time or on the evolution of the model in former. times

We can also differ the mathematical models according to kind (character) of the quantities and variables figuring in the model. First division is into

- ◇ **deterministic** - there is no random quantity in the model,
- ◇ **stochastic** - we can find at least one random quantity in the model.

Second, and for us the most important, is following

- ◇ **continuous** - the quantities take the values from some continuum,
- ◇ **discrete** - the quantities take at most countable values.

2.3 The System Identification Procedure

In this section our goal is to describe the system identification loop. Especially, three basic entities, validation of the model and the reasons why the model can be deficient.

2.3.1 Three basic items

The construction of a model from input-output data involves three basic items:

A data set. The input and output data are often recorded during a specifically designed identification experiment, where the user can determine which signals to measure, when to measure them and may also choose the input signals. The objective with experiment design is to make these choices so that the data will be maximally informative and illuminative, exposed to constraints that may be at hand.

A set of candidate models or the model structure. A set of candidate models is obtained by specifying in which collection of models we are going to look for a suitable one. This is without a doubt the most important and also the most difficult choice of the system identification procedure.
Sometimes the set of candidate models is obtained after careful modeling. Then a model with some unknown physical parameters is constructed from basic laws of physics and other well-established relationships. This sort of model sets may be called *gray box*.
In other cases linear models may be employed without mentioning of the physical background. Such a model set, which parameters do not reflect physical consideration in the system, is called *black box*.

Determining the "best" model in the set of candidate models, guided by the data. In this entity we want determine the rule by which candidate models can be assessed using the data. The assessment of the quality of the model is typically based on how the models perform when they attempt to reproduce or refresh the measured data.

2.3.2 Validation of the model

After our discussion on three basic entities and their clarifying, the next question arises: Is the chosen model "good enough"? Which means, whether it is valid for its purpose or not? This problem can be answered after some testing of this model. Such tests are known as model validation and they involve various procedure to establish how the model relates to observed data, to prior knowledge, and to its intended use. Deficient model behavior in these aspects make us reject the model, while good construction of the model will develop a certain confidence in it. A model can never be accepted as a final and as a true description of the system. Rather, a model can at best be regarded as a "good enough" performance of certain respects that are of particular interest to us.

2.3.3 The System Identification Loop

The system identification procedure which we described in two previous sections has a logical flow: first collect data, then choose a model set, then select the "best" model in this set, that best describes the data according to the chosen criterion (see figure 2.1). It is quite probable, that the model first obtained will not pass the validation process. This is the reason why we must go back and revise the various steps of the procedure. Let us denote the reasons for which the model may be deficient.

- ◇ The set of input-output data was not illuminative and informative enough to provide guidance in assorting suitable models.
- ◇ The criterion was not well chosen.

- ◇ The numerical procedure failed to find the best model according to our criterion.
- ◇ The set of candidate models was not appropriate, in that it did not contain any "good enough" description of the system.

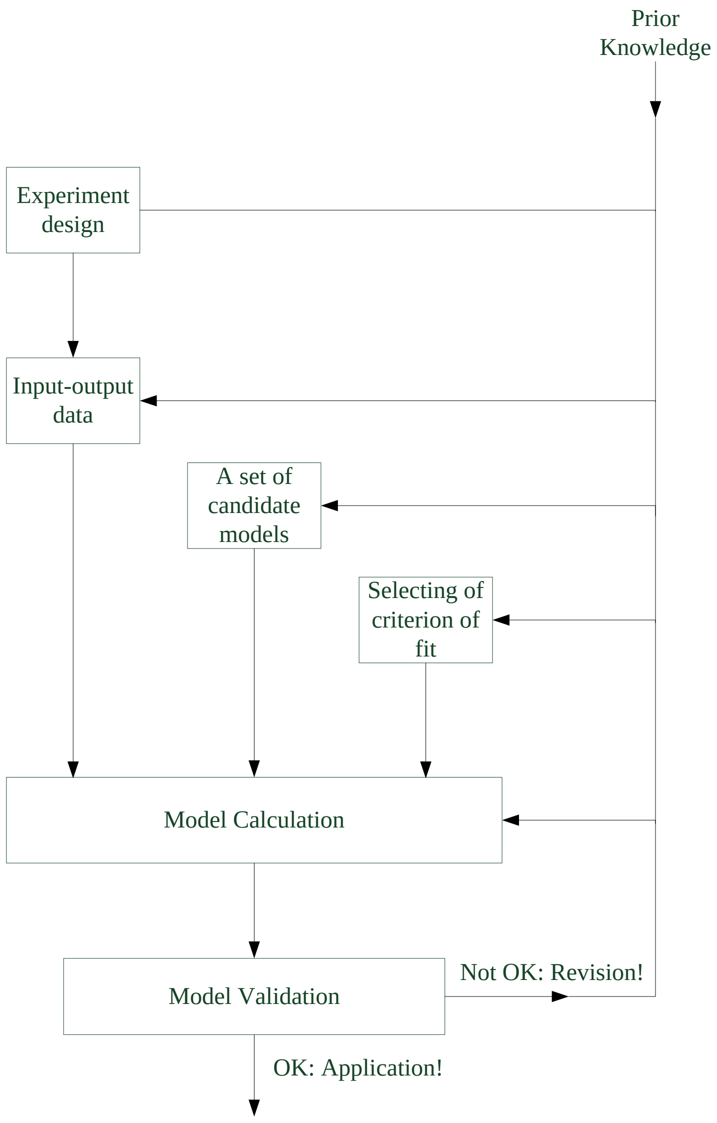

Figure 2.1: The system identification loop

Chapter 3

Linear Dynamic Systems

In the previous chapter we classified the dynamic system into two principal groups. The first one was **continuous** dynamic systems and the second one was **discrete** dynamic systems. In this chapter we will describe both of these systems.

It is known the mathematical description of the dynamic system is divided into two basic groups, namely **external** and **internal** description.

External description is the expression of the dynamic properties with the aid of the relation between the input and output variables. This description does not provide us with the information about the internal states of the system. By measuring of the input and output magnitudes we can obtain only the external description of the system.

Internal description is the relation between the input variables, the states of the system, and the output variables. Then we are talking about the **state equations** of the system.

However, we have to also remind that the dynamic system consists of the state space and the dynamic conditions. Due to this fact, we shall introduce auxiliary state vector $\boldsymbol{x}(t)$. In the state-space form the relationship between the input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$ signals is represented by a system of first-order differential or difference equations using this auxiliary state vector $\boldsymbol{x}(t)$.

In the following table you can see the general mathematical models of continuous and discrete dynamic systems both in time invariant and time variant form.

GENERAL	Time-Invariant	Time-Variant
Continuous	$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$	$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$
Discrete	$\boldsymbol{x}_i(t) = \boldsymbol{f}(\boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1})$	$\boldsymbol{x}_i(t) = \boldsymbol{f}(i, \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1})$

Table 3.1: General Mathematical Models of Dynamic Systems

3.1 Continuous-time Models

For most physical systems it is suitable and easier to construct models with physical insight in continuous time than in discrete time, simply because most of physical laws (e.g. Newton's law of motion) are expressed in continuous time.

At first we shall modify our first table 3.1. We will focus on the continuous part and will extend it by the addition of linear model.

CONTINUOUS	Time-Invariant	Time-Variant
General	$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$	$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$
Linear	$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t)$	$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t)$

Table 3.2: Continuous Mathematical Models of Dynamic Systems

As we can see the modeling of the dynamic of system normally leads to a representation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t) \tag{3.1}$$

where

- $\boldsymbol{x}(t)$ is a $r \times 1$ **state vector**,
- $\boldsymbol{F}(t)$ is a $r \times r$ **matrix of the system** (sometimes so called a matrix of the dynamic coefficients),
- $\boldsymbol{u}(t)$ is a $m \times 1$ **input vector**,
- $\boldsymbol{G}(t)$ is a $r \times m$ **matrix interconnecting the input with the state of the system**.

Notice that both matrices $\boldsymbol{F}(t)$ and $\boldsymbol{G}(t)$ depend on time. We can thus talk about **time--varying** systems. Further we have to realize that the state variables are not measurable in contrast to inputs and outputs and we must deduce them from the input variables. We have to express the relationship between all three vectors of the dynamic system by the following so called **measurement equation**, again in matrix notation

$$\boldsymbol{y}(t) = \boldsymbol{H}(t)\boldsymbol{x}(t) + \boldsymbol{C}(t)\boldsymbol{u}(t) \tag{3.2}$$

where

- $\boldsymbol{y}(t)$ is a $n \times 1$ **output vector** or **measurement vector**,
- $\boldsymbol{H}(t)$ is a $n \times r$ **matrix of the measuring sensitivity**,
- $\boldsymbol{C}(t)$ is a $n \times m$ **matrix interconnecting the input with the output**.

Both matrices $\boldsymbol{H}(t)$, $\mathbf{C}(t)$ are known functions of the time.

Finally, the model of the continuous linear dynamic system is formed by the equations (3.1) and (3.2), i.e. by the state equation and the measurement equation, respectively.

The next question naturally arises: How to solve the state equation (3.1)? The first way is to solve it by standard technique for solving of the differential equations. Another way is to use the Laplace transform.

The solution of the (3.1) obtained by the standard technique is following

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau. \tag{3.3}$$

We can note that in the previous expression figures the matrix $\Phi(t, \tau)$. This matrix represents a **state transition matrix** (because it transforms the solution at time t into the solution at time τ) and holds

$$\boldsymbol{\Phi}(\tau, t) = \boldsymbol{\Phi}(\tau)\boldsymbol{\Phi}^{-1}(t)$$

where $\boldsymbol{\Phi}(\cdot)$ is the **fundamental matrix**. The fundamental matrix is the solution of

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t), \quad t \in \langle 0, T \rangle, \tag{3.4}$$

i.e. the homogeneous part of the state equation (3.1), and satisfies

$$\dot{\boldsymbol{\Phi}}(t) = \boldsymbol{F}(t)\boldsymbol{\Phi}(t), \qquad \boldsymbol{\Phi}(0) = \boldsymbol{I} \tag{3.5}$$

where $\boldsymbol{I}$ is an identity matrix of the dimension r. Then the homogeneous solution implying from two previous formulations has form

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t)\boldsymbol{x}(0).$$

This expression means that matrix $\boldsymbol{\Phi}(t)$ transform the initial state $\boldsymbol{x}(0)$ of the dynamic system into the corresponding state $\boldsymbol{x}(t)$ of the dynamic system at time t.

If the elements of the matrix $\boldsymbol{F}(t)$ are continuous functions on the given time interval then we have guaranteed existence and regularity of the fundamental matrix $\boldsymbol{\Phi}(t)$. If we start from the presumption that the fundamental matrix is regular then it is possible to write

$$\boldsymbol{\Phi}^{-1}(t)\boldsymbol{x}(t) = \boldsymbol{x}(0), \qquad \boldsymbol{\Phi}(\tau)\boldsymbol{\Phi}^{-1}(t)\boldsymbol{x}(t) = \boldsymbol{x}(\tau). \tag{3.6}$$

Evidently, the second formula implies from the first one. If we substitute τ instead of t into the first expression and equate them we obtain the second formalization.

Properties of the state transition matrix:

$$\begin{array}{ll} \diamond\ \boldsymbol{\Phi}(\tau, 0) = \boldsymbol{\Phi}(\tau), & \diamond\ \boldsymbol{\Phi}(t_2, t_1)\boldsymbol{\Phi}(t_1, t_0) = \boldsymbol{\Phi}(t_2, t_0), \\ \diamond\ \boldsymbol{\Phi}(\tau, \tau) = \boldsymbol{\Phi}(0) = \boldsymbol{I}, & \diamond\ \dfrac{\partial \boldsymbol{\Phi}(\tau, t)}{\partial \tau} = \boldsymbol{F}(t)\boldsymbol{\Phi}(\tau, t). \\ \diamond\ \boldsymbol{\Phi}(\tau, t) = \boldsymbol{\Phi}^{-1}(t, \tau), & \end{array} \tag{3.7}$$

By using $\boldsymbol{\Phi}(\tau, t) = \boldsymbol{\Phi}(\tau)\boldsymbol{\Phi}^{-1}(t)$ we can modify the solution of the state equation (3.3) in the following way

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t)\boldsymbol{\Phi}^{-1}(t_0)\boldsymbol{x}(t_0) + \boldsymbol{\Phi}(t)\int_{t_0}^{t} \boldsymbol{\Phi}^{-1}(\tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau. \tag{3.8}$$

Two cases can occur:

$\boldsymbol{F}(t) \equiv \boldsymbol{F}$ The elements of the matrix of dynamic coefficients are constant. Keep in mind that in this case we are talking about time-invariant system and thus the matrix $\boldsymbol{\Phi}(t, \tau)$ depends only on the difference $t - \tau$ and we can write

$$\begin{aligned}\boldsymbol{x}(t) &= \mathrm{e}^{\boldsymbol{F}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} \mathrm{e}^{\boldsymbol{F}(t-\tau)}\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau = \\ &= \mathrm{e}^{\boldsymbol{F}(t-t_0)}\boldsymbol{x}(t_0) + \mathrm{e}^{\boldsymbol{F}(t)}\int_{t_0}^{t} \mathrm{e}^{\boldsymbol{F}(-\tau)}\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau\end{aligned}$$

where

$$\boldsymbol{\Phi(t, \tau)} = \mathrm{e}^{\boldsymbol{F}(t-\tau)} = \sum_{k=0}^{\infty} \frac{\boldsymbol{F}^k (t-\tau)^k}{k!}.$$

The term $\mathrm{e}^{\boldsymbol{F}t}$, appearing in the above formulations, can be solved by the Laplace transform

$$\boldsymbol{\Phi}(t) = \mathrm{e}^{\boldsymbol{F}t} = \mathcal{L}^{-1}\left(\frac{1}{s\boldsymbol{I} - \boldsymbol{F}}\right)$$

where $\mathcal{L}^{-1}$ is the operator of the inverse Laplace transform, s is the Laplace variable, and $\boldsymbol{I}$ denotes the identity matrix.

General case In the time-variant case, there are many different functions that may satisfy the requirements (3.5), (3.7), and the solution is dependent on the structure of the dynamic system. We can use the numerical methods to obtain the solution of the homogeneous differential equation (3.4) such as Euler method or Backward Euler. But it is more convenient to apply higher order methods on this problem, for example Runge-Kutta methods.

3.2 Discrete-time Models

Discretization concerns the process of transferring continuous models and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on computers. Discretization is often performed within the meaning of the forward difference

$$\Delta\boldsymbol{x}(t_i) = \boldsymbol{x}(t_{i+1}) - \boldsymbol{x}(t_i),$$

expressed as a function of all independent variables (including the value $\boldsymbol{x}(t_i)$ as the independent variable)

$$\boldsymbol{x}(t_{i+1}) = f(t_i, \boldsymbol{x}(t_i), \boldsymbol{u}(t_i)).$$

For the linear case of discrete-time dynamic system we can express this functional dependence in matrix notation

$$\boldsymbol{x}(t_{i+1}) = \boldsymbol{F}(t_i)\boldsymbol{x}(t_i) + \boldsymbol{G}(t_i)\boldsymbol{u}(t_i). \tag{3.9}$$

By analogy, we obtain by discretization of (3.2) the discrete measurement equation

$$\boldsymbol{y}(t_i) = \boldsymbol{H}(t_i)\boldsymbol{x}(t_i) + \boldsymbol{C}(t_i)\boldsymbol{u}(t_i).$$

The solution of the discrete state equation (3.9) is

$$\boldsymbol{x}(t_i) = \boldsymbol{\Phi}(t_i, t_{i-1})\boldsymbol{x}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \boldsymbol{\Phi}(t_i, \tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau \tag{3.10}$$

where $\boldsymbol{\Phi}(t_i, t_{i-1})$ denotes the state transition matrix for the linear discrete-time dynamic system.

If the value of the output $\boldsymbol{u}(\tau) \equiv \boldsymbol{u}$ at time interval $\langle t_{i-1}, t_i \rangle$ then we can rewrite the solution as

$$\boldsymbol{x}(t_i) = \boldsymbol{\Phi}(t_i, t_{i-1})\boldsymbol{x}(t_{i-1}) + \boldsymbol{\Gamma}(t_{i-1})\boldsymbol{u}(t_{i-1}) \tag{3.11}$$

with

$$\boldsymbol{\Gamma}(t_{i-1}) = \int_{t_{i-1}}^{t_i} \boldsymbol{\Phi}(t_i, \tau)\boldsymbol{G}(\tau)\,\mathrm{d}\tau.$$

This solution has with the aid of short notation

$$\boldsymbol{x}_i \equiv \boldsymbol{x}(t_i), \quad \boldsymbol{u}_i \equiv \boldsymbol{u}(t_i), \quad \boldsymbol{y}_i \equiv \boldsymbol{y}(t_i),$$

$$\boldsymbol{\Phi}_{i-1} \equiv \boldsymbol{\Phi}(t_i, t_{i-1}), \quad \boldsymbol{\Gamma}_i \equiv \boldsymbol{\Gamma}(t_i), \quad \boldsymbol{H}_i \equiv \boldsymbol{H}(t_i), \quad \boldsymbol{C}_i \equiv \boldsymbol{C}(t_i)$$

the following compact form

$$\boldsymbol{x}_i = \boldsymbol{\Phi}_{i-1}\boldsymbol{x}_{i-1} + \boldsymbol{\Gamma}_{i-1}\boldsymbol{u}_{i-1}.$$

Finally, by modifying the introductory table 3.1 as before in the continuous-time models, we obtain the following table 3.3 for the discrete-time models.

DISCRETE	Time-Invariant	Time-Variant
General	$\boldsymbol{x_i} = \boldsymbol{f}(\boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1})$	$\boldsymbol{x}_i = \boldsymbol{f}(i, \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1})$
Linear	$\boldsymbol{x}_i = \boldsymbol{\Phi}\boldsymbol{x}_{i-1} + \boldsymbol{\Gamma}\boldsymbol{u}_{i-1}$	$\boldsymbol{x}_i = \boldsymbol{\Phi}_{i-1}\boldsymbol{x}_{i-1} + \boldsymbol{\Gamma}_{i-1}\boldsymbol{u}_{i-1}$

Table 3.3: Discrete Mathematical Models of Dynamic Systems

Chapter 4

Stochastic Systems

This chapter deals with a stochastic systems where states and inputs are random processes. A random (or stochastic) process is one whose behavior is non-deterministic, in that a subsequent state of the system is determined both by the predictable actions of the process and by a random element. The indeterminacy in a future evolution of a stochastic process is described by probability distributions hence we shall place here needful notes from probability theory. The second part of this chapter will be engaged in a very short analysis of the linear stochastic systems. See [3], [4] and [10] for any more information.

4.1 Short View into Probability Theory

Some fundamentals of probability theory which are necessary to gain insight into the recursive algorithm shall sum up in this chapter. These concepts will also be instrumental in describing random phenomena and properties of stochastic systems. The presentation will be intentionally brief, assuming that the reader has already some general knowledge with the subject.

4.1.1 Characteristics of Random Variable

In this subsection we introduce the characteristics of random variables. These results are relevant for two reasons. First, since random variables can be found in almost any practical application, it is important to be able to understand and manipulate random variables. The second reason is that the characterization of random variables will serve for our development of random processes in two following subsections.

1. **Expected Value**

$$\mathrm{E}(X) = \begin{cases} \sum\limits_{k=0}^{\infty} x_k P(X = x_k) & \text{if } X \text{ is discrete random variable,} \\ \\ \int\limits_{-\infty}^{\infty} x f(x)\,\mathrm{d}x & \text{if } X \text{ is continious random variable} \end{cases}$$

   where $P(X = x_k)$ and $f(x)$ denotes probability function and density function, respectively.

2. **Variance**
$$\mathrm{Var}(X) = \mathrm{E}[(X - \mathrm{E}(X))^2] = \mathrm{E}(X^2) - \mathrm{E}^2(X) = \sigma_X^2.$$
The square root of the variance, σ_X, is known as **standard deviation**.

3. **Covariance**
$$\mathrm{Cov}(X, Y) = \mathrm{E}[(X - \mathrm{E}(X))(Y - \mathrm{E}(Y))] = \mathrm{E}(XY) - \mu_X \mu_Y = c_{XY}.$$
where $\mu_X = \mathrm{E}(X)$ and $\mu_Y = \mathrm{E}(Y)$ are the **means** (or expected values) of random variables X, Y, respectively.

4. **Correlation**
$$r_{XY} = \mathrm{E}(XY).$$

5. **Correlation Coefficient**
$$\rho_{XY} = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}} = \frac{r_{XY} - \mu_X \mu_Y}{\sigma_X \sigma_Y}, \qquad |\rho_{XY}| \leq 1.$$

Theorem 4.1. *Two random variables that satisfy* $\mathrm{E}(XY) = \mathrm{E}(X)\mathrm{E}(Y)$ *are said to be* ***uncorrelated****.*

In other words, X and Y will be uncorrelated if their covariance is zero, $c_{XY} = 0$.

Theorem 4.2. *Two random variables X and Y are said to be* ***orthogonal*** *if their correlation is zero, $r_{XY} = 0$.*

Normal (Gaussian) Random Variables $X \sim N(\mu, \sigma^2)$

Gaussian random variables play a central role in probability theory. A random variable, X, taking values in $(-\infty, \infty)$ with the following parameters $\mu \in (-\infty, \infty)$, and $\sigma^2 \in (0, \infty)$, is said to be **Gaussian** if its probability density function has form
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right\} \tag{4.1}$$
where μ and σ^2 are the mean and the variance, respectively. Such a normal random variable is denoted by $N(\mu, \sigma^2)$. The $N(0, 1)$ random variable is called **standard** normal random variable.

4.1.2 Parameter Estimation: Bias

Consider the problem of estimating the value of the parameter, $\boldsymbol{\theta}$, from a sequence of random variables X_n, for $n = 1, \ldots, N$. Since the estimate is a function of N random variables, we will denote it by $\hat{\boldsymbol{\theta}}_N$. In general we would like the estimate to be equal, in the sense of the expectation, to the true value. The difference between the expected value of the estimate and actual value, $\boldsymbol{\theta}$, is called the **bias**, denoted by $\mathcal{B}$,
$$\mathcal{B} = \boldsymbol{\theta} - \mathrm{E}(\hat{\boldsymbol{\theta}}_N).$$

If the bias is zero, then the expectation of the estimate is equal to the true value

$$\mathrm{E}(\hat{\boldsymbol{\theta}}_N) = \boldsymbol{\theta}$$

and the estimate is said to be **unbiased**. If $\mathcal{B} \neq 0$, then $\hat{\boldsymbol{\theta}}_N$ is said to be **biased**. If an estimate is biased but the bias goes to zero as the number of observations, N, goes to infinity

$$\lim_{N\to\infty} \mathrm{E}(\hat{\boldsymbol{\theta}}_N) = \boldsymbol{\theta}$$

then the estimate is said to be **asymptotically unbiased**. In general, it is covetable that an estimator be either unbiased or asymptotically unbiased.

4.1.3 Stochastic (Random) Processes and Their Characteristics

We shall deal with the characterization and analysis of random processes. Since a random process is simply an indexed sequence of the random variables, we shall just extend the concepts from the part 4.1.1 to random processes. In the second part of this subsection we will summarize the characteristics of the stochastic processes.

Definition 4.1. A random process is a sequence of random variables $\{X_t; t \in \mathcal{T}\}$ or $\{X_t\}_{t\in\mathcal{T}}$, where $\mathcal{T}$ is an index set, defined on a common probability space $(\Omega, \mathcal{B}, P)$.

Also in this case the random processes are divided into discrete-time and continuous--time stochastic processes, i.e.

◇ $\mathcal{T} \subseteq \mathbb{R}$ - continuous-time random process,

◇ $\mathcal{T} \subseteq \mathbb{Z}$ - discrete-time random process.

Characteristics of Random Process
Consider two stochastic processes $\{X_t\}_{t\in\mathcal{T}}$, and $\{Y_t\}_{t\in\mathcal{T}}$ both defined on the same probability space.

1. **Expected value** $\mu_X(t) = \mathrm{E}(X_t)$.
2. **Variance** $\sigma_X^2(t) = \mathrm{Var}(X_t) = \mathrm{Cov}(X_t, X_t)$.
3. **Autocovariance** $c_X(t_1, t_2) = \mathrm{Cov}(X_{t_1}, X_{t_2})$.
4. **Autocorrelation** $r_X(t_1, t_2) = \mathrm{E}(X_{t_1}, X_{t_2})$.
5. **Cross-Covariance** $c_{XY}(t_1, t_2) = \mathrm{Cov}(X_{t_1}, Y_{t_2})$.
6. **Cross-Correlation** $r_{XY}(t_1, t_2) = \mathrm{E}(X_{t_1}, Y_{t_2})$.
7. **Autocorrelation Matrix**
   Consider $\boldsymbol{X} = [X_1, X_2, \ldots, X_d]^T$ is a vector of d values of a process X_n, then the outer product

$$\boldsymbol{X}\boldsymbol{X}^T = \begin{bmatrix} X_1X_1 & X_1X_2 & \cdots & X_1X_d \\ X_2X_1 & X_2X_2 & \cdots & X_2X_d \\ \vdots & \vdots & \ddots & \vdots \\ X_dX_1 & X_dX_2 & \cdots & X_dX_d \end{bmatrix}$$

is a $d \times d$ matrix. The corresponding autocorrelation matrix of the dimension $d \times d$, using $r_X(p) = r_X(-p)$, has the form

$$\boldsymbol{R_X} = \mathrm{E}(\boldsymbol{X}\boldsymbol{X}^T) \begin{bmatrix} r_X(0) & r_X(1) & \cdots & r_X(d-1) \\ r_X(1) & r_X(0) & \cdots & r_X(d-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_X(d-1) & r_X(d-2) & \cdots & r_X(0) \end{bmatrix}.$$

8. **Autocovariance Matrix** $\quad \boldsymbol{C_X} = \boldsymbol{R_X} - \boldsymbol{\mu_X}\boldsymbol{\mu_X}^T$
   where $\boldsymbol{\mu_X} = [\mu_{\boldsymbol{X}}, \mu_{\boldsymbol{X}}, \dots \mu_{\boldsymbol{X}}]^T$ is a vector of length d containing the mean value of the process.

Theorem 4.3. *Two random processes X_t and Y_t satisfying $c_{XY}(t_1, t_2) = 0$ for all t_1 and t_2 are said to be **uncorrelated**.*

Theorem 4.4. *Two random processes X_t and Y_t are said to be **orthogonal** if their cross-correlation is zero, $r_{XY}(t_1, t_2) = 0$, for all t_1 and t_2.*

Theorem 4.5. *A continuous random process X_t is said to be **uncorrelated** if holds*

$$c_X(t_1, t_2) = Q(t_1, t_2)\delta(t_1 - t_2)$$

where $\delta(t)$ is a unit impulse function (or the Dirac's delta)

$$\delta(t) = \begin{cases} \infty, & t = 0, \\ 0, & t \neq 0, \end{cases} \qquad \int_{-\infty}^{\infty} \delta(t)\,\mathrm{d}t = 1.$$

Similar relation holds for a discrete random process, however for this case the unit impulse function is replaced by unit sample function (or the Kronecker delta) $\Delta(i)$, i.e.

$$\Delta(i) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0. \end{cases}$$

Definition 4.2. A random process X_t is said to be **white** (or **white noise**) if its mean and autocovariance function satisfy the following

$$\mu_X(t) = \mu, \qquad c_X(t_1, t_2) = \begin{cases} \sigma^2, & t_1 = t_2, \\ 0, & k \neq l, \end{cases}$$

i.e. if it is a constant-mean process and its values X_{t_1} and X_{t_2} are uncorrelated random variables for every t_1 and t_2, each having a variance σ_X^2.

4.2 Linear Stochastic Systems

As we said at the beginning of this chapter the word "stochastic" means "pertaining to chance", and is thus used to describe subjects that contain some element of random (or stochastic) behavior. For a system to be stochastic, one or more parts of the system has

CONTINUOUS	DISCRETE
$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t) + \boldsymbol{v}(t)$	$\boldsymbol{x_i} = \boldsymbol{\Phi}_{i-1}\boldsymbol{x}_{i-1} + \boldsymbol{\Gamma}_{i-1}\boldsymbol{u}_{i-1} + \boldsymbol{v}_{i-1}$
$\boldsymbol{y}(t) = \boldsymbol{H}(t)\boldsymbol{x}(t) + \boldsymbol{C}(t)\boldsymbol{u}(t) + \boldsymbol{w}(t)$	$\boldsymbol{y}_i = \boldsymbol{H}_i\boldsymbol{x}_i + \boldsymbol{C}_i\boldsymbol{u}_i + \boldsymbol{w}_i$

Table 4.1: Continuous and Discrete Mathematical Models of Linear Stochastic Systems

randomness associated with it. Unlike a deterministic system which we considered in the chapter 3, for example, a stochastic system does not always produce the same output for a given input. The table 4.1 contains both continuous and discrete time mathematical models of the linear stochastic systems which we obtained from the deterministic models by implementation of another random inputs to the system.

By reason of simplicity we shall consider that the deterministic input vector $\boldsymbol{u}$, and the stationary random processes $\boldsymbol{v}$, $\boldsymbol{w}$ called the noise of process and measurement, respectively, have the same probability distribution and are independent on the previous values of the state and input. Both noises we shall consider as white, i.e. with these following properties

- ◇ **continuous-time case**

$$\mathrm{E}\left(\begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{w}(t) \end{bmatrix}\right) = 0, \quad \mathrm{Cov}\left(\begin{bmatrix} \boldsymbol{v}(t_1) \\ \boldsymbol{w}(t_1) \end{bmatrix}, \begin{bmatrix} \boldsymbol{v}(t_2) \\ \boldsymbol{w}(t_2) \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & \boldsymbol{R} \end{bmatrix} \delta(t_2 - t_1),$$

- ◇ **discrete-time case**

$$\mathrm{E}\left(\begin{bmatrix} \boldsymbol{v}_{i-1} \\ \boldsymbol{w}_i \end{bmatrix}\right) = 0, \quad \mathrm{Cov}\left(\begin{bmatrix} \boldsymbol{v}_{i-1} \\ \boldsymbol{w}_{i-1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{v}_j \\ \boldsymbol{w}_j \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & \boldsymbol{R} \end{bmatrix} \Delta(i - 1 - j).$$

4.2.1 Continuous-time Models

We shall study the linear continuous-time stochastic system modeled by equations

$$\begin{array}{rcccccc} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{F}(t)\boldsymbol{x}(t) & + & \boldsymbol{G}(t)\boldsymbol{u}(t) & + & \boldsymbol{v}(t), \\ \boldsymbol{y}(t) & = & \boldsymbol{H}(t)\boldsymbol{x}(t) & + & \boldsymbol{C}(t)\boldsymbol{u}(t) & + & \boldsymbol{w}(t). \end{array} \tag{4.2}$$

The state equation can be expressed writing $\frac{\mathrm{d}}{\mathrm{d}t}$ instead of the dot

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t) + \frac{\mathrm{d}\boldsymbol{b}(t)}{\mathrm{d}t}$$

where $\boldsymbol{b}(t)$ is called **Wiener process** or **Brownian motion**. Finally, multiplying by "$\mathrm{d}t$"

$$\mathrm{d}\boldsymbol{x}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t)\mathrm{d}t + \boldsymbol{G}(t)\boldsymbol{u}(t)\mathrm{d}t + \mathrm{d}\boldsymbol{b}(t) \tag{4.3}$$

we obtained the system of linear stochastic differential equations. The increment of the Wiener process $\mathrm{d}\boldsymbol{b}(t)$ has the zero mean, covariance

$$\mathrm{Cov}(\mathrm{d}\boldsymbol{b}(t), \mathrm{d}\boldsymbol{b}(t)) = \mathrm{E}(\mathrm{d}\boldsymbol{b}(t)\mathrm{d}\boldsymbol{b}^T(t)) = \boldsymbol{Q}\mathrm{d}t,$$

and the values of increments in nonoverlaping time intervals are independent.
The derivation of the Wiener process

$$\boldsymbol{v}(t) = \frac{\mathrm{d}\boldsymbol{b}(t)}{\mathrm{d}t}$$

does not exist. The exact definition of the continuous-time stochastic system therefore is much more complicated than in the case of the discrete-time. This problem is analyzed in detail in [2].

The solution of the state equation in (4.2) is determined by the following formula

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t,t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{v}(\tau)\,\mathrm{d}\tau. \tag{4.4}$$

Now we shall focus on the evolution of the expectation and covariance of the state and output vector of the system (4.2). Taking into account that the expectation of white noises $\boldsymbol{v}(t)$, $\boldsymbol{w}(t)$ is equal to zero we find by virtue of (4.4) and the measurement equation in (4.2) the following formulas for the mean of the system state and output vector, $\boldsymbol{x}(t)$, and $\boldsymbol{y}(t)$, respectively,

$$\mu_{\boldsymbol{x}}(t) = \boldsymbol{\Phi}(t,t_0)\mu_{\boldsymbol{x}}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\,\mathrm{d}\tau,$$

$$\mu_{\boldsymbol{y}}(t) = \boldsymbol{H}(t)\mu_{\boldsymbol{x}}(t) + \boldsymbol{C}(t)\boldsymbol{u}(t).$$

As we can see the expectation value of the state and output vector of the stochastic system evolves as well as in the case of deterministic system.

The covariance function of the state vector is given by the formula

$$c_{\boldsymbol{x}}(t_1,t_2) = \boldsymbol{\Phi}(t_1,t_0)c_{\boldsymbol{x}}(t_0,t_0)\boldsymbol{\Phi}(t_2,t_0) + \int_{t_0}^{\min(t_1,t_2)} \boldsymbol{\Phi}(t_1,\tau)\boldsymbol{Q}\boldsymbol{\Phi}(t_2,\tau)\,\mathrm{d}\tau.$$

While deriving this expression we remembered that the initial state $\boldsymbol{x}(t_0)$ is independent of the white noise $\boldsymbol{v}(t)$ at $t > t_0$ and that $\boldsymbol{\Phi}(t,\tau) = 0$ at $\tau > t$. Due to this the upper limit of integration is equal to $\min(t_1,t_2)$.

4.2.2 Discrete-time Models

The linear discrete-time stochastic system is described by the following model

$$\begin{array}{ccccccc} \boldsymbol{x}_i & = & \boldsymbol{\Phi}_{i-1}\boldsymbol{x}_{i-1} & + & \boldsymbol{\Gamma}_{i-1}\boldsymbol{u}_{i-1} & + & \boldsymbol{v}_{i-1}, \\ \boldsymbol{y}_i & = & \boldsymbol{H}_i\boldsymbol{x}_i & + & \boldsymbol{C}_i\boldsymbol{u}_i & + & \boldsymbol{w}_i. \end{array} \tag{4.5}$$

Much like in the continuous-time models of the linear stochastic systems the evolution of the expectation of the state and output vector is same for both deterministic and stochastic systems

$$\mathrm{E}(\boldsymbol{x}_i) = \boldsymbol{\Phi}_{i-1}\mathrm{E}(\boldsymbol{x}_{i-1}) + \boldsymbol{\Gamma}_{i-1}\boldsymbol{u}_{i-1},$$

$$\mathrm{E}(\boldsymbol{y}_i) = \boldsymbol{H}_i\mathrm{E}(\boldsymbol{x}_i) + \boldsymbol{C}_i\boldsymbol{u}_i.$$

For the evolution of the covariance function of the state we use the relation determining the mean error of the state

$$\widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \mathrm{E}(\boldsymbol{x}_i) \quad \Rightarrow \quad \mathrm{E}(\boldsymbol{x}_i) = \boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i. \tag{4.6}$$

By application of (4.6) into the expectation of the system state and output vector and by deduction of (4.5) we get

$$\begin{aligned} \widetilde{\boldsymbol{x}}_i &= \boldsymbol{\Phi}_{i-1}\widetilde{\boldsymbol{x}}_{i-1} + \boldsymbol{v}_{i-1}, \\ \widetilde{\boldsymbol{y}}_i &= \boldsymbol{H}_i\widetilde{\boldsymbol{x}}_i + \boldsymbol{w}_i, \end{aligned}$$

and so

$$\widetilde{\boldsymbol{x}}_i\widetilde{\boldsymbol{x}}_i^T = \boldsymbol{\Phi}_{i-1}\widetilde{\boldsymbol{x}}_{i-1}\widetilde{\boldsymbol{x}}_{i-1}^T\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{\Phi}_{i-1}\widetilde{\boldsymbol{x}}_{i-1}\boldsymbol{v}_{i-1}^T + \boldsymbol{v}_{i-1}\widetilde{\boldsymbol{x}}_{i-1}^T\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{v}_{i-1}\boldsymbol{v}_{i-1}^T.$$

Again, taking into account that the expectation of white noises $\boldsymbol{v}_{i-1}$, $\boldsymbol{w}_i$ is equal to zero we obtain the expression for the evolution of the covariance function the following formula

$$\mathrm{Cov}(\boldsymbol{x}_i, \boldsymbol{x}_i) = \mathrm{E}(\widetilde{\boldsymbol{x}}_i\widetilde{\boldsymbol{x}}_i^T) = \boldsymbol{\Phi}_{i-1}\mathrm{Cov}(\boldsymbol{x}_{i-1}, \boldsymbol{x}_{i-1})\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{Q}.$$

Now we define for the covariance matrix this denotation

$$\boldsymbol{K}_{\boldsymbol{x}_i} = \mathrm{Cov}(\boldsymbol{x}_i, \boldsymbol{x}_i)$$

and overwrite the previous equation into the form

$$\boldsymbol{K}_{x_i} = \boldsymbol{\Phi}_{i-1}\boldsymbol{K}_{\boldsymbol{x}_{i-1}}\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{Q}.$$

In the same way it is possible to derive the relations for the joint covariance matrix of the system state and output vector

$$\mathrm{Cov}\left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{\Phi}_{i-1}\boldsymbol{K}_{\boldsymbol{x}_{i-1}}\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{Q} & \boldsymbol{\Phi}_{i-1}\boldsymbol{K}_{\boldsymbol{x}_{i-1}}\boldsymbol{H}_i^T + \boldsymbol{S} \\ \boldsymbol{H}_i\boldsymbol{K}_{\boldsymbol{x}_{i-1}}\boldsymbol{\Phi}_{i-1}^T + \boldsymbol{S}^T & \boldsymbol{H}_i\boldsymbol{K}_{\boldsymbol{x}_{i-1}}\boldsymbol{H}_i^T + \boldsymbol{R} \end{bmatrix}.$$

If we consider both white noises $\boldsymbol{v}$, and $\boldsymbol{w}$ as the Gaussian processes and the initial state of the system with the Gaussian distribution, i.e. $\boldsymbol{x}_0 \sim N(\mu, \sigma^2)$, then also the system state and output are the Gaussian processes fully characterized by equations for the evolution of the expectation value and joint covariance matrix.

Chapter 5

Parameter Estimation Methods

We are now in the situation that we have selected a certain model structure $\mathcal{M}$, with particular models $\mathcal{M}(\theta)$ parameterized using the parameter vector $\boldsymbol{\theta} \in \mathcal{D}_{\mathcal{M}} \subset \mathbb{R}^d$. Then the set of models is

$$\mathcal{M}^\star = \{\mathcal{M}(\boldsymbol{\theta}) | \boldsymbol{\theta} \in \mathcal{D}_{\mathcal{M}}\}.$$

We have also collected the batch of data of the inputs and outputs from the system over a time interval $1 \le t \le N$

$$Z^N = \{y(1), \boldsymbol{u}(1), y(2), \boldsymbol{u}(2) \dots, y(N), \boldsymbol{u}(N)\}.$$

The problem we are faced with is to resolve upon how to use the information contained in Z^N to select a proper value $\hat{\boldsymbol{\theta}}_N$ of the parameter vector, and hence a proper member $\mathcal{M}(\hat{\boldsymbol{\theta}})$ in the set $\mathcal{M}^\star$. More precise, we have to determine a mapping from the data set Z^N to the set $\mathcal{D}_{\mathcal{M}}$

$$Z^N \rightarrow \hat{\boldsymbol{\theta}}_N \in \mathcal{D}_{\mathcal{M}}. \tag{5.1}$$

Such a mapping is a **parameter estimation method**.

We used some lines above the vector $\hat{\boldsymbol{\theta}}_N$. Now we should explain what indicates. The estimate $\hat{\boldsymbol{\theta}}_N$ is defined by minimization of the function of the model parameter $\boldsymbol{\theta}$

$$V_N(\boldsymbol{\theta}, Z^N), \tag{5.2}$$

for given Z^N, i.e.

$$\hat{\boldsymbol{\theta}}_N = \hat{\boldsymbol{\theta}}_N(Z^N) = \arg \min_{\boldsymbol{\theta} \in \mathcal{D}_{\mathcal{M}}} V_N(\boldsymbol{\theta}, Z^N). \tag{5.3}$$

Here the "arg min" means the minimizing argument of the function, i.e. that value of $\boldsymbol{\theta}$ which minimizes V_N. If the minimum is not unique, we let arg min denote the set of minimizing arguments. The mapping (5.1) is thus defined implicitly by (5.3). The sources [8] and [11] were worked up for taking in writing the sections of this chapter.

5.1 Linear Least Square Method (LLS)

The predictor

$$\hat{y}(t|\boldsymbol{\theta}) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} = \boldsymbol{\theta}^T\boldsymbol{\varphi}(t)$$

is a scalar product between a known data vector $\boldsymbol{\varphi}(t)$ and the parameter vector $\boldsymbol{\theta}$. Such a model is called a linear regression in statistics (linear in parameter $\boldsymbol{\theta}$). Recall the vector $\boldsymbol{\varphi}(t)$ is known as the vector of regressors (chapter 2). It is of importance since powerful and simple estimation methods can be applied for the determination of $\boldsymbol{\theta}$. We shall discuss the linear least squares method for the estimation of $\boldsymbol{\theta}$ in this section.

An approach is to select $\boldsymbol{\theta}$ in (1.1) so as to fit the calculated values $\hat{y}(t|\boldsymbol{\theta})$ as well as possible to the measured outputs by the least squares method

$$\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta}, Z^N)$$

where

$$V_N(\boldsymbol{\theta}, Z^N) = \sum_{t=1}^{N} \varepsilon(t)^2 = \sum_{t=1}^{N} \left(y(t) - \hat{y}(t|\boldsymbol{\theta})\right)^2 = \sum_{t=1}^{N} \left(y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}\right)^2. \tag{5.4}$$

Since V_N is quadratic in $\boldsymbol{\theta}$, we can find the minimum value easily by setting the derivative to zero

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} V_N(\boldsymbol{\theta}, Z^N) = 2\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\left(y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}\right) = 0$$

which gives

$$\sum_{t=1}^{N} \boldsymbol{\varphi}(t)y(t) = \sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{\theta}$$

or

$$\hat{\boldsymbol{\theta}}_N^{LLS} = \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right]^{-1} \sum_{t=1}^{N} \boldsymbol{\varphi}(t)y(t). \tag{5.5}$$

Once the vectors $\boldsymbol{\varphi}(t)$ are known, or defined, the solution can be easily found by numerical software, such as MATLAB.

The estimation (5.5) can be rewritten as

$$\hat{\boldsymbol{\theta}}_N^{LLS} = \boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(N)\mathbf{r}_{\boldsymbol{\varphi}y}(N) \tag{5.6}$$

with

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(N) = \sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t), \tag{5.7}$$

so called a $d \times d$ deterministic **autocorrelation matrix**, and

$$\boldsymbol{r}_{\boldsymbol{\varphi}y}(N) = \sum_{t=1}^{N} \boldsymbol{\varphi}(t)y(t), \tag{5.8}$$

so called a $d \times 1$ deterministic **vector of cross-correlation** between $y(t)$ and $\boldsymbol{\varphi}(t)$.

An alternative way how to view $\hat{\boldsymbol{\theta}}_N^{LLS}$ is as the solution of

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(N)\hat{\boldsymbol{\theta}}_N^{LLS} = \boldsymbol{r}_{\boldsymbol{\varphi}y}(N). \tag{5.9}$$

These last equations are known as the **normal equations**. If the matrix of autocorrelation $\boldsymbol{R}_{\boldsymbol{\varphi}}$ is positive definite, i.e. for any vector $\boldsymbol{z}$ must be satisfied $\boldsymbol{z}^T\boldsymbol{R}_{\boldsymbol{\varphi}}\boldsymbol{z} > 0$, then the solution of the normal equations $\hat{\boldsymbol{\theta}}_N^{LLS}$ is a unique solution of the minimization problem

$$\hat{\boldsymbol{\theta}}_N^{LLS} = \arg\min_{\boldsymbol{\theta}\in\mathcal{D}_\mathcal{M}} V_N(\boldsymbol{\theta}, Z^N) = \arg\min_{\boldsymbol{\theta}\in\mathcal{D}_\mathcal{M}} \sum_{t=1}^{N}(y(t)-\boldsymbol{\varphi}^T(t)\boldsymbol{\theta})^2.$$

5.2 Nonlinear Least Squares Method (NLS)

The nonlinear least squares method is the form of the least squares analysis which is used to fit a set of observations with a model that is non-linear in unknown parameters. It is used in some forms of non-linear regression. The basis of the method is to approximate the model by a linear one and to refine the parameters by successive iterations. There are many similarities to linear least squares, but also some significant differences.

In the nonlinear case the predictor is given by the nonlinear equation (nonlinear in $\boldsymbol{\theta}$)

$$\hat{y}(t|\boldsymbol{\theta}) = g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}) \tag{5.10}$$

where the parameter $\boldsymbol{\theta}$ needs to be estimated. Subsequently, the corresponding prediction error is

$$\varepsilon(t) = y(t) - g(\boldsymbol{\varphi}(\boldsymbol{t}), \boldsymbol{\theta}).$$

The principle is same as in the previous section. The criterion function V_N has the form

$$V_N(\boldsymbol{\theta}, Z^N) = \sum_{t=1}^{N}\varepsilon(t)^2 = \sum_{t=1}^{N}\left(y(t)-\hat{y}(t|\boldsymbol{\theta})\right)^2 = \sum_{t=1}^{N}\left(y(t)-g(\boldsymbol{\varphi}(t),\boldsymbol{\theta})\right)^2. \tag{5.11}$$

The minimum value of $V_N(\boldsymbol{\theta}, Z^N)$ occurs when the gradient is zero

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}V_N(\boldsymbol{\theta}, Z^N) = 2\sum_{t=1}^{N}\varepsilon(t)\frac{\mathrm{d}\varepsilon(t)}{\mathrm{d}\boldsymbol{\theta}} = 0. \tag{5.12}$$

These gradient equations do not have a closed solution. The above nonlinear minimization problem can be solved by locally linearizing the function $g(\boldsymbol{\varphi}(t), \boldsymbol{\theta})$ and iteratively solving for the parameter $\boldsymbol{\theta}$ which minimizes (5.11).

The initial value must be chosen for the parameter $\boldsymbol{\theta}$. Then, the parameter is refined iteratively, which means, the value is obtained by successive approximation

$$\boldsymbol{\theta} \approx \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta_t} + \triangle\boldsymbol{\theta}.$$

The vector of increments, $\triangle\boldsymbol{\theta}$, is known as the shift vector. At each iteration the model is linearized by approximation to a first-order Taylor series expansion about $\boldsymbol{\theta}_t$

$$g(\boldsymbol{\varphi}(t), \boldsymbol{\theta} \approx g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t) + \frac{\partial g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t)}{\partial\boldsymbol{\theta}}\triangle\boldsymbol{\theta} = g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t) + Dg|_{\boldsymbol{\theta}_t}\triangle\boldsymbol{\theta}.$$

In terms of the linearized model

$$\frac{\partial\varepsilon(t)}{\partial\boldsymbol{\theta}} = -\frac{\partial g(\boldsymbol{\varphi}(t), \boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = -Dg|_{\boldsymbol{\theta}_t}$$

and the residuals are given by

$$\varepsilon(t) = y(t) - g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t) - Dg|_{\boldsymbol{\theta}_t} \triangle\boldsymbol{\theta} = \triangle y(t) - Dg|_{\boldsymbol{\theta}_t}.$$

Substituting these expressions into gradient equations (5.12), they become

$$-2 \sum_{t=1}^{N} Dg|_{\boldsymbol{\theta}_t} (\triangle y(t) - Dg|_{\boldsymbol{\theta}_t} \triangle\boldsymbol{\theta}) = 0.$$

After a rearrangement we obtain again the normal equations

$$\sum_{t=1}^{N} Dg|_{\boldsymbol{\theta}_t} Dg|_{\boldsymbol{\theta}_t} \triangle\theta = \sum_{t=1}^{N} Dg|_{\boldsymbol{\theta}_t} \triangle y(t).$$

Finally, for the vector of increments, $\triangle\boldsymbol{\theta}$, holds

$$\triangle\hat{\boldsymbol{\theta}}^{NLS} = Dg|_{\boldsymbol{\theta}_t}{}^{\dagger} \triangle y(t) = Dg|_{\boldsymbol{\theta}_t}{}^{\dagger} (y(t) - g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t))$$

with $Dg|_{\boldsymbol{\theta}_t}{}^{\dagger}$ as linear least square pseudo inverse of $Dg|_{\boldsymbol{\theta}_t}$.

Each iteration step will be then

$$\hat{\boldsymbol{\theta}}_{t+1}^{NLS} = \hat{\boldsymbol{\theta}}_{t}^{NLS} + Dg|_{\boldsymbol{\theta}_t}{}^{\dagger} (y(t) - g(\boldsymbol{\varphi}(t), \boldsymbol{\theta}_t)). \tag{5.13}$$

Equation (5.13) is initialized with the initial value, $\hat{\boldsymbol{\theta}}_0^{NLS}$, and is iterated until the solution converges.

Differences between linear and nonlinear least squares

◇ Many solution algorithms for NLS require the initial value for the parameter, LLS does not.

◇ Many solution algorithms for NLS require that the gradient be calculated. Analytical expressions for partial derivatives can be complicated. If analytical expressions are impossible to obtain the partial derivatives must be calculated by numerical approximation.

◇ In NLS non-convergence (non-convergence means failure of the algorithm to find a minimum) is a common phenomenon whereas the LLS is globally concave and so non-convergence is not a question at issue.

◇ NLS is usually an iterative process. The iterative process has to be terminated when a convergence criterion is satisfied. LLS solutions can be computed using direct methods.

◇ In LLS the solution is unique, but in NLS there may be multiple minima in the sum of squares.

5.3 Weighted Least Squares Method (WLS)

In most of modeling applications it may not be reasonable to assume that every observation should be treated equally (because some observations are more important or more precise than the others). Weighted least squares can often be used to maximize the efficiency of parameter estimation. This is done by giving each observed data its proper amount of influence by assignment of the weight over the parameter estimates.

Weighted least squares reflects the behavior of the random errors in the model, and it can be used with functions that are either linear or nonlinear in the parameter $\boldsymbol{\theta}$. It works by incorporating extra nonnegative constants (weights) associated with each observed data, into the criterion function. The size of the weight indicates the precision of the information contained in the associated observation.

The weighted least square criterion for the linear predictor takes the form

$$V_N(\boldsymbol{\theta}, Z^N) = \sum_{t=1}^{N} \beta(N,t)\varepsilon(t)^2 = \sum_{t=1}^{N} \beta(N,t)\left(y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}\right)^2 \tag{5.14}$$

where $\beta(N,t)$ is the weight. Then the expression for the resulting estimate is quite analogous to (5.5)

$$\hat{\boldsymbol{\theta}}_N^{WLS} = \left[\sum_{t=1}^{N} \beta(N,t)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right]^{-1} \sum_{t=1}^{N} \beta(N,t)\boldsymbol{\varphi}(t)y(t). \tag{5.15}$$

Chapter 6

Adaptive Filtering

Filter is often described as device in the form of a piece hardware or software applied to a set of noisy data in order to extract information about a prescribed quantity of interest. The term **filtering** thus means the extraction of information about a quantity of interest at time t by using data measured up to and including time t. By the **adaptive filtering** we mean the device that is **self-designing** in that the adaptive filter relies on a recursive algorithm. That makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics (a priori statistical information) is not available.

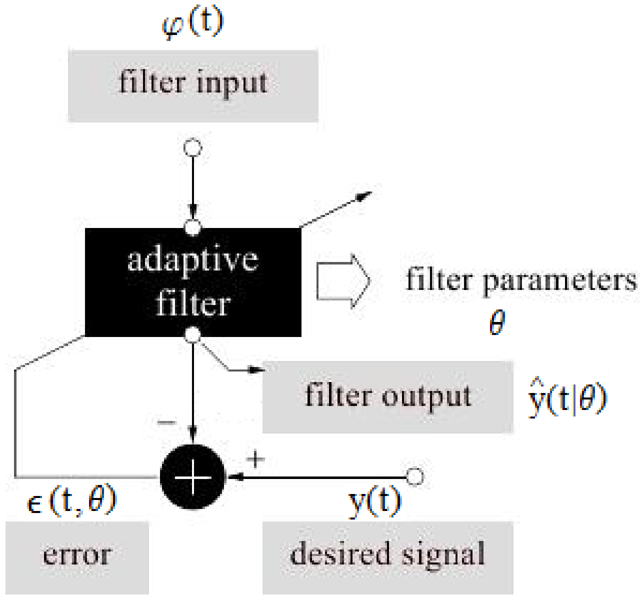

Figure 6.1: Adaptive filter

The picture figures that basic operation involves two processes in adaptive filter,namely a filtering process and adaptation process. The adaptation process includes adjusting filter parameters to time-varying environment. In other words, the adaptation process is updating the filter when new data arrives. The adaptation is controlled by error signal.

The following section explains the recursive least squares method, which is based on the weighted least squares method. For finding more details about the adaptive filters and recursive least squares algorithm use [5] and [8].

6.1 The Recursive Least Squares Method (RLS)

The recursive least squares algorithm is one of the most popular adaptive algorithms and can be easily and exactly derived from the normal equations. This algorithm may be viewed as a special case of the Kalman filter. Our main mission in this section is to develop the basic theory of the recursive least squares algorithm as an important tool for linear adaptive filtering.

Let us consider the filter coefficients $\boldsymbol{\theta}_t$ as our subject of interest that minimize, at time t, the weighted least squares criterion

$$V_t(\boldsymbol{\theta}, Z^t) = \sum_{i=1}^{t} \beta(t,i)\varepsilon(i)^2 = \sum_{i=1}^{t} \beta(t,i)\left(y(i) - \boldsymbol{\varphi}^T(i)\boldsymbol{\theta}\right)^2. \tag{6.1}$$

Here, we introduced

$$Z^t = (y^t, \boldsymbol{u}^t) = \{y(1), \boldsymbol{u}(1), \ldots, y(t), \boldsymbol{u}(t)\}$$

to denote the input-output measurements available at time t.

The weighting factor $\beta(t,i)$ in equation (6.1) has the property

$$0 < \beta(t,i) \leq 1, \qquad i = 1, 2, \ldots, t.$$

The use of the weighting factor $\beta(t,i)$, in general, is intended to ensure that the data in the distant past are "forgotten". A special form of weighting that is commonly used is the **exponential weighting factor** known also as **forgetting factor** defined by

$$\beta(t,i) = \lambda^{t-i}, \qquad i = 1, 2, \ldots, t$$

where λ is positive constant close to, but less than, 1. When λ equals one, we have the ordinary method of least squares.

By using the forgetting factor we obtain the exponentially weighted least squares criterion

$$V_t(\boldsymbol{\theta}, Z^t) = \sum_{i=1}^{t} \lambda^{t-i}\left(y(i) - \boldsymbol{\varphi}^T(i)\boldsymbol{\theta}\right)^2. \tag{6.2}$$

The estimated value of the vector of filter coefficients is defined with using (5.15) by

$$\hat{\boldsymbol{\theta}}_t^{RLS} = \left[\sum_{i=1}^{t} \lambda^{t-i}\boldsymbol{\varphi}(i)\boldsymbol{\varphi}^T(i)\right]^{-1} \sum_{i=1}^{t} \lambda^{t-i}\boldsymbol{\varphi}(i)y(i). \tag{6.3}$$

Again, the typical way how to see the equation (6.3) is the form given by the normal equations, in matrix notation

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(t)\hat{\boldsymbol{\theta}}_t^{RLS} = \boldsymbol{r}_{\boldsymbol{\varphi}y}(t). \tag{6.4}$$

The $d \times d$ deterministic autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$ is now defined by

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(t) = \sum_{i=1}^{t} \lambda^{t-i} \boldsymbol{\varphi}(i) \boldsymbol{\varphi}^T(i). \tag{6.5}$$

The $d \times 1$ deterministic cross-correlation vector $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$ between the regression vector and the desired signal is correspondingly defined by

$$\boldsymbol{r}_{\boldsymbol{\varphi}y}(t) = \sum_{i=1}^{t} \lambda^{t-i} \boldsymbol{\varphi}(i) y(i). \tag{6.6}$$

For easier notation we set

$$\hat{\boldsymbol{\theta}}_t^{RLS} = \hat{\boldsymbol{\theta}}_t.$$

6.1.1 Recursive Algorithm

Since $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$ and $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$ both depend on time t, instead of solving the deterministic normal equations directly, for each value of t, we will derive a recursive solution of the form

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \triangle \hat{\boldsymbol{\theta}}_{t-1}$$

where $\triangle \hat{\boldsymbol{\theta}}_{t-1}$ is a correction that is applied to the solution at time $t-1$. Because

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t) \boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$$

the recursion will be derived first by expressing $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$, in terms of $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1)$, and then by deriving a recursion that allows us to evaluate $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$ in terms of $\boldsymbol{R}_{\boldsymbol{\varphi}}(t-1)$.

By isolating the term corresponding to $i = t$ from the rest of the summation on the right-hand side of the formalization (6.6), we can write

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{\varphi}y}(t) &= \sum_{i=1}^{t} \lambda^{t-i} \boldsymbol{\varphi}(i) y(i) = \\ &= \sum_{i=1}^{t-1} \lambda^{t-i} \boldsymbol{\varphi}(i) y(i) + \lambda^0 \boldsymbol{\varphi}(t) \boldsymbol{\varphi}^T(t) = \\ &= \lambda \left[\sum_{i=1}^{t} \lambda^{t-1-i} \boldsymbol{\varphi}(i) y(i) \right] + \boldsymbol{\varphi}(t) y(t). \end{aligned} \tag{6.7}$$

However, by definition, the expression inside the square brackets on the right-hand side of (6.7) equals the cross-correlation vector $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1)$. Hence, we have the following recursion for updating the value of the cross-correlation vector

$$\boldsymbol{r}_{\boldsymbol{\varphi}y}(t) = \lambda \boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1) + \boldsymbol{\varphi}(t) y(t) \tag{6.8}$$

where $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1)$ is the "old" value of the vector of cross-correlation, and the product $\boldsymbol{\varphi}(t) y(t)$ plays the role of a correction term in the updating operation. Similarly, the autocorrelation matrix may be updated from $\boldsymbol{R}_{\boldsymbol{\varphi}}(t-1)$ and new input-output data vector $\boldsymbol{\varphi}(t)$ using the recursion

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(t) = \lambda \boldsymbol{R}_{\boldsymbol{\varphi}}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t). \tag{6.9}$$

However, seeing that it is the inverse of $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$ in which we are interested, we may apply **Woodbury's Identity** to the expression (6.9) to get the desired recursion. Woodbury's identity is also referred to in the standard literature as the **Matrix Inversion Lemma** and is expressed as

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v})^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

Taking $\boldsymbol{A} = \lambda \boldsymbol{R}_{\boldsymbol{\varphi}}(t-1)$, $\boldsymbol{u} = \boldsymbol{\varphi}(t)$, and $\boldsymbol{v} = \boldsymbol{\varphi}^T(t)$ we obtain the following recursion for the inverse of $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$

$$\boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t) = \lambda^{-1}\boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t-1) - \frac{\lambda^{-2}\boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t-1)}{1 + \lambda^{-1}\boldsymbol{\varphi}^T(t)\boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t-1)\boldsymbol{\varphi}(t)}. \tag{6.10}$$

To simplify last notation, we will let $\boldsymbol{P}(t)$ denote the inverse of autocorrelation matrix at time t,

$$\boldsymbol{P}(t) = \boldsymbol{R}_{\boldsymbol{\varphi}}^{-1}(t) \tag{6.11}$$

and determine what is referred to as the **gain vector**, $\boldsymbol{g}(t)$, as follows

$$\boldsymbol{g}(t) = \frac{\lambda^{-1}\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}{1 + \lambda^{-1}\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)} = \frac{\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda + \boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}. \tag{6.12}$$

Incorporating the formulas (6.11), and (6.12) to the expression (6.10) gives

$$\boldsymbol{P}(t) = \lambda^{-1}\left[\boldsymbol{P}(t-1) - \boldsymbol{g}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\right]. \tag{6.13}$$

An useful expression for the gain vector may be derived from formalization (6.12) hereby, cross-multiplying to eliminate the denominator on the right side of equation (6.12) we obtain

$$\boldsymbol{g}(t) + \lambda^{-1}\boldsymbol{g}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t) = \lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{\varphi}(t).$$

Next, bringing the second term on the left to the right side of the equation leads to

$$\boldsymbol{g}(t) = \lambda^{-1}\left[\boldsymbol{P}(n-1) - \boldsymbol{g}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\right]\boldsymbol{\varphi}(t).$$

Finally, from expression (6.13) we can see that the term multiplying $\boldsymbol{\varphi}(t)$ is $\boldsymbol{P}(t)$ and we have

$$\boldsymbol{g}(t) = \boldsymbol{P}(t)\boldsymbol{\varphi}(t). \tag{6.14}$$

The gain vector is thus the solution to the normal equations

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(t)\boldsymbol{g}(t) = \boldsymbol{\varphi}(t).$$

To complete the recursion, we must derive the time update equation for the vector of filter coefficients $\hat{\boldsymbol{\theta}}$. With

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{P}(t)\boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$$

it follows form the update for $\boldsymbol{r}_{\boldsymbol{\varphi}y}(t)$ (6.8) defined in (6.6) that

$$\hat{\boldsymbol{\theta}}_t = \lambda \boldsymbol{P}(t)\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1) + \boldsymbol{P}(t)\boldsymbol{\varphi}(t)y(t). \tag{6.15}$$

By incorporating the update for $\boldsymbol{P}(t)$ given by (6.13) into the first term on the right hand side of the last equation and substituting (6.14) we obtain

$$\hat{\boldsymbol{\theta}}_t = \left[\boldsymbol{P}(t-1) - \boldsymbol{g}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\right]\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1) + \boldsymbol{g}(t)y(t).$$

Finally, by recognizing that $\boldsymbol{P}(t-1)\boldsymbol{r}_{\boldsymbol{\varphi}y}(t-1) = \hat{\boldsymbol{\theta}}_{t-1}$ we obtain the recursion

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{g}(t)\left[y(t) - \boldsymbol{\varphi}(t)\hat{\boldsymbol{\theta}}_{t-1}\right] \tag{6.16}$$

which can be written

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \xi(t)\boldsymbol{g}(t) \tag{6.17}$$

with

$$\xi(t) = y(t) - \boldsymbol{\varphi}(t)^T\hat{\boldsymbol{\theta}}_{t-1} \tag{6.18}$$

the difference between $y(t)$ and its estimation that is formed by applying the previous vector of filter coefficients, $\boldsymbol{\theta}_{t-1}$, to the new data vector $\boldsymbol{\varphi}(t)$. This sequence, called the **a priori error** is the error that would occur if the filter coefficients were not updated. On the other hand, the **a posteriori error** is the error that occurs after the vector of filter coefficients is updated, which means

$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}(t)^T\hat{\boldsymbol{\theta}}_t.$$

Parameters:	$d =$ Filter order
	$\lambda =$ Forgetting factor
	$\delta =$ Value used to initialize $\boldsymbol{P}(0)$
Initialization:	$\boldsymbol{\theta}_0 = 0$
	$\boldsymbol{P}(0) = \delta^{-1}\boldsymbol{I}$
Computation:	For $t = 1, 2, \ldots$
	$\boldsymbol{g}(t) = \dfrac{\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda + \boldsymbol{\varphi}(t)^T\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}$
	$\xi(t) = y(t) - \boldsymbol{\varphi}(t)^T\hat{\boldsymbol{\theta}}_{t-1}$
	$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \xi(t)\boldsymbol{g}(t)$
	$\boldsymbol{P}(t) = \lambda^{-1}\left[\boldsymbol{P}(t-1) - \boldsymbol{g}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{P}(t-1)\right]$

Table 6.1: The Recursive Least Squares Algorithm

6.1.2 Initialization of RLS Algorithm

This subsection concerns the initialization of the RLS algorithm. Since the recursive algorithm given by the table 6.1 involves the recursive updating of the vector of filter coefficients $\hat{\boldsymbol{\theta}}_t$ and the inverse autocorrelation matrix $\boldsymbol{P}(t)$, initial conditions for both of these terms are required.

There are two ways how this initialization is typically performed. The first is to build up the autocorrelation matrix recursively according to (6.9) and then evaluate the inverse directly

$$\boldsymbol{P}(0) = \left[\sum_{i=-t_0}^{0} \lambda^{-i} \boldsymbol{\varphi}(i) \boldsymbol{\varphi}^T(i) \right]^{-1}.$$

The regression vector $\boldsymbol{\varphi}(i)$ is obtained from an initial block of data for $-t_0 \leq i \leq 0$. Computing the cross-correlation vector $\boldsymbol{r}_{\boldsymbol{\varphi} y}(0)$ in the same manner,

$$\boldsymbol{r}_{\boldsymbol{\varphi} y}(0) = \sum_{i=-t_0}^{0} \lambda^{-i} \boldsymbol{\varphi}(i) y(i),$$

we may then initialize $\hat{\boldsymbol{\theta}}_0$ by setting $\hat{\boldsymbol{\theta}}_0 = \boldsymbol{P}(0)\boldsymbol{r}_{\boldsymbol{\varphi} y}(0)$.

A simpler approach is to modify the expression slightly for the autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$ by writing

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(t) = \sum_{i=1}^{t} \lambda^{t-i} \boldsymbol{\varphi}(i) \boldsymbol{\varphi}^T(i) + \delta \lambda^t \boldsymbol{I}$$

where $\boldsymbol{I}$ is the $d \times d$ identity matrix, and δ is a small positive constant. Thus putting $t = 0$ in the last expression, we have

$$\boldsymbol{R}_{\boldsymbol{\varphi}}(0) = \delta \boldsymbol{I}.$$

Correspondingly, for the initial value of $\boldsymbol{P}(t)$ equal to the inverse of the correlation matrix $\boldsymbol{R}_{\boldsymbol{\varphi}}(t)$, we set

$$\boldsymbol{P}(0) = \delta^{-1} \boldsymbol{I}. \tag{6.19}$$

It only remains for us to choose an initial value for the vector of filter coefficients. It is common to set

$$\hat{\boldsymbol{\theta}}_0 = \mathbf{0} \tag{6.20}$$

where $\mathbf{0}$ is the $d \times 1$ null vector.

The initialization procedure incorporating equations (6.19), (6.20) is referred to as a **soft-constrained initialization** and was used in our simulations. The positive constant δ is the only parameter required for this initialization and this fact is judged as the advantage. The disadvantage of this approach is that it introduce a bias in the least squares solution. However, with an exponential weighting factor, $\lambda < 1$, this bias go to zero as t increases.

6.1.3 Choice of the Forgetting Factor

As we said before, we need to select the forgetting factor so that criterion essentially contains those measurements that are relevant for the current properties of the system.

For a system that changes gradually, the most customary choice is to take a constant forgetting factor

$$\lambda(t) \equiv \lambda.$$

The constant λ is always chosen slightly less than 1 and we can write

$$\lambda^{t-i} = \mathrm{e}^{(t-i)\log\lambda} \approx \mathrm{e}^{-(t-i)(1-\lambda)}.$$

This means that measurements that are older than inverse of $1-\lambda$ samples are included in the criterion with a weight that is $\mathrm{e}^{-1} \approx 36\%$ of that of the most recent measurement. Roughly speaking,

$$T = \frac{1}{1-\lambda} \tag{6.21}$$

is a measure of the memory of the algorithm. If the system remains approximately constant over T samples, a suitable choice of λ can then be made from (6.21). Typical choices of λ are in the range between $0,98$ and $0,995$.

For a system that undergoes abrupt and sudden changes, an adaptive choice of λ could be conceived. When a sudden change of the system has been detected, it is suitable to decrease $\lambda(t)$ to a small value for one sample, thereby "cutting off" past measurements from the criterion, and then to increase $\lambda(t)$ to a value close to 1 again.

Chapter 7

Modeling of Vehicle Dynamics

For the theoretical analysis of vehicle dynamics, the equations of motion have to be known and the physical interactions between the various subsystems have to be written in the form of mathematical equations. Two main approaches can be considered for the construction of the vehicle model. If we want to obtain a model which is exact and precise as possible, we use the methods of theoretical physics such as Lagrange or Euler. The alternative approach is to attempt to model the vehicle as simply as possible and with as little computing-time as possible. For this occasion emphasis is placed here on the classical **single-track model** (also known as a **bicycle model**).

In this chapter we shall introduce two models, which are commonly used for vehicle dynamics control. First we shall shortly deal with linear tire model which describes the tire performance during a motion of a vehicle. Consequently we shall target bicycle model by which the fundamental concept of vehicle lateral dynamics is explained.

7.1 Tire Model

The primary forces during lateral maneuvering, acceleration, and braking are generated by tires. The linear tire model doesn't consider longitudinal tire forces, hence it is suitable for analyzing a stable vehicle behavior under the assumption of small steering and acceleration.

The lateral forces, which serve for control the direction of the vehicle and are generated by a tire, are a function of the slip angle. The slip angle, α, represents the angle containing the velocity vector of wheel displacement (in another words, the tire's direction of travel) with its longitudinal axis.

As the tire rolls, the tire contact patch over the ground deflects by slip angle α and deforms according to the direction of travel (see figure 7.1). This deformation and the elasticity of the tire produce in the tire contact patch the lateral tire force. The dependence of the lateral tire force on the slip angle is investigated experimentally. In the linear region of the tire curve ($0° \leq \alpha \leq 3°$) the lateral force may be expressed by the following

$$F_y = -C_\alpha \alpha \tag{7.1}$$

where the cornering stiffness, C_α, represents the slope of initial part of the tire curve displayed in the figure 7.2.

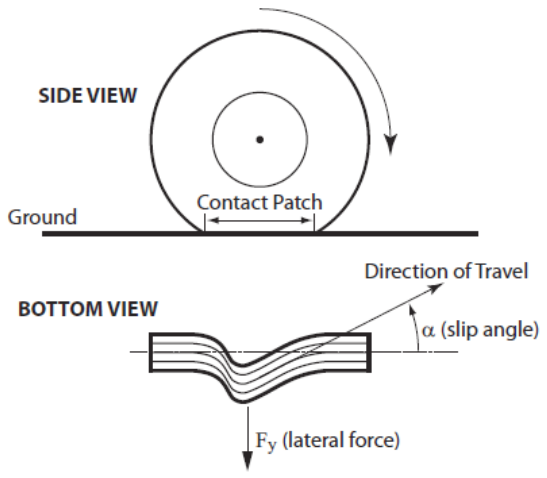

Figure 7.1: The rolling tire deformation and the lateral force

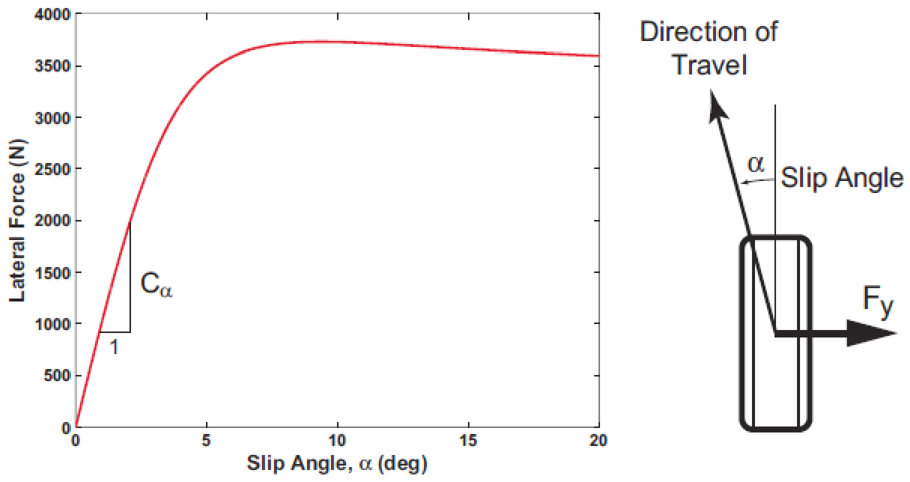

Figure 7.2: Tire lateral force and tire slip angle

This topic is more in detail described in [7] or [12] with tire self-aligning torque. For our purpose this presented simplified model is sufficient.

7.2 Single-Track Model

We shall inquire into the vehicle **steerability**, i.e. the steering response at the constant driving velocity. The figure 7.3 shows the vehicle-driver-road control loop, where the steering response (the output of the schema) can be for example the yaw rate or the lateral velocity. Next figure 7.4 illustrate for more clear visualization the vehicle in co-ordinate system which has its origin at the vehicle center of gravity.

We shall theoretically investigate the lateral dynamics of the vehicle in the horizontal plane. For this purpose the single-track model, or bicycle model is most often used representation. This model gives good results for non-critical driving situation and is

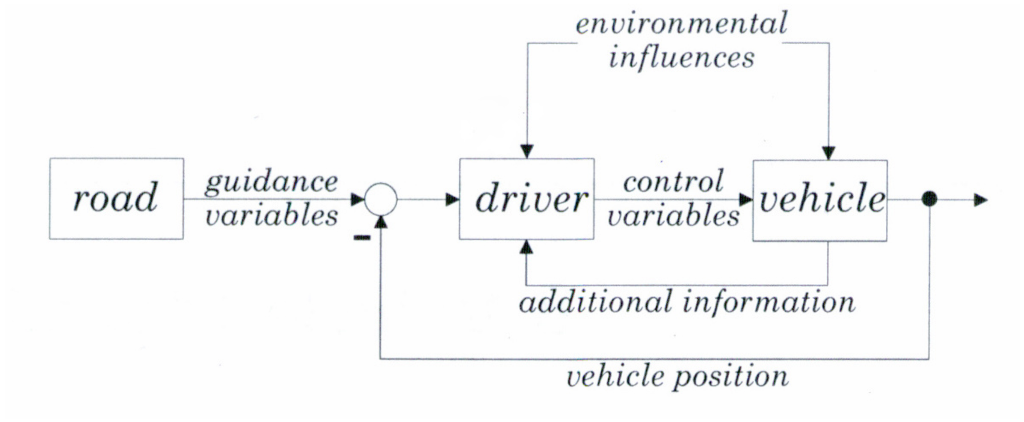

Figure 7.3: The standard vehicle-driver-road control loop

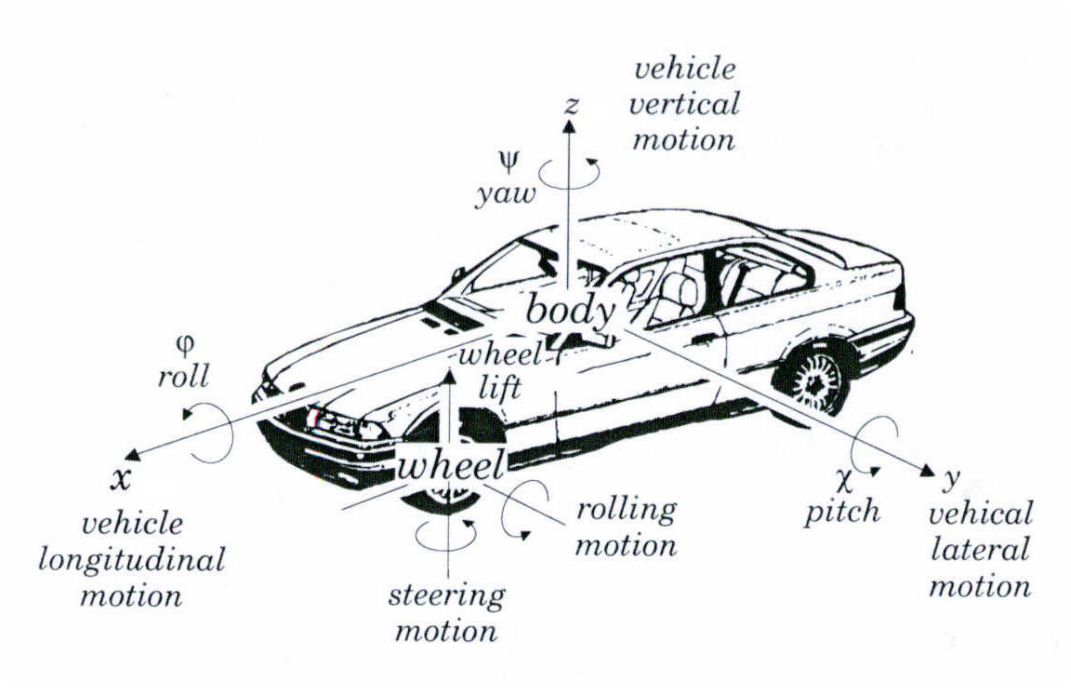

Figure 7.4: The vehicle in coordinate system

frequently applied for many test maneuvers. The first publication of this modeling dates back to 1940. The states looming in this representation are states of the lateral velocity and yaw rate. Detailed derivation and explanation can be found in many books, for example in [12].

In the bicycle model the wheels on each axis are considered as a single unit. Because of this, it is only possible to derive the one single tire slip angle for the left and right wheels on the front axle, α_f, and one for the wheels on the rear axle. The approximation of angles

$$\sin\omega \approx \omega, \qquad \cos\omega \approx 1 \tag{7.2}$$

is applied to linearize the problem. The center of gravity (CG) of the vehicle lies at a plane of a carriageway (or at road-level) hence we can vanish the suspension roll. On this account we shall assume that the tire slip angles on the inside and outside wheels are approximately the same. For this planar bicycle model we shall not investigate the effect of the side wind, the influence of the rear steering and we shall also consider absolutely stiff steering.

In figure 7.5, $F_{y,f}$ and $F_{y,r}$ are the front and rear lateral tire forces, respectively, α_f and α_r are corresponding tire slip angles, u_x and u_y are the longitudinal and the lateral components of the vehicle velocity, and δ is the steering angle.

The following force and moment balances are a basis for derivation of the equation of motion for the bicycle model

$$\begin{aligned} ma_y &= F_{y,f}\cos\delta + F_{y,r}, \\ I_z\dot{r} &= aF_{y,f}\cos\delta - bF_{y,r} \end{aligned} \tag{7.3}$$

where I_z is the yaw moment of inertia of the vehicle, m is the vehicle mass, a and b are the distance of the front and rear axles from the center of gravity, and a_y is the lateral vehicle acceleration.

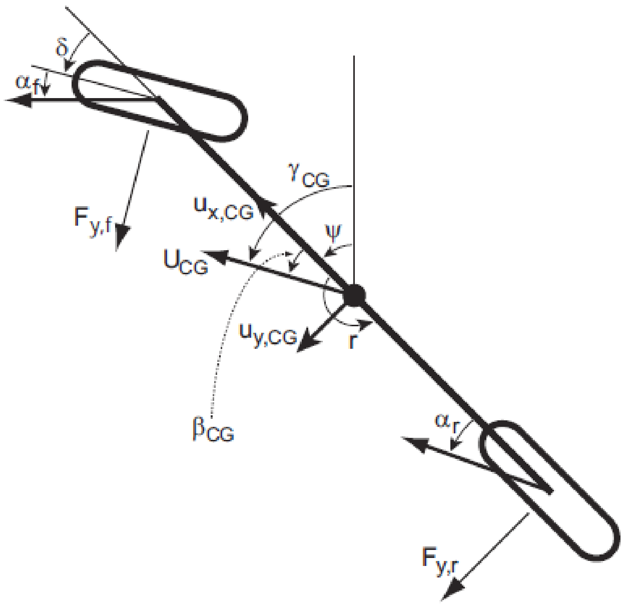

Figure 7.5: The single-track model

The state equation for the bicycle model can be written in the form

$$\begin{bmatrix} \dot{u}_{y,CG} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{-C_{\alpha_f}-C_{\alpha_r}}{mu_{x,CG}} & -u_{x,CG} - \frac{C_{\alpha_f}a-C_{\alpha_r}b}{mu_{x,CG}} \\ \frac{-C_{\alpha_f}a-C_{\alpha_r}b}{I_z u_{x,CG}} & \frac{-C_{\alpha_f}a^2-C_{\alpha_r}b^2}{I_z u_{x,CG}} \end{bmatrix} \begin{bmatrix} u_{y,CG} \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{mi_r} \\ \frac{C_{\alpha_f}a}{I_z i_r} \end{bmatrix} \delta \tag{7.4}$$

where i_r denotes the steering ratio.

For the front and the rear tire forces, $F_{y,f}$ and $F_{y,r}$, respectively, by using the linear tire model (7.1), hold formulas

$$F_{y,f} = -C_{\alpha_f}\alpha_f, \qquad F_{y,r} = -C_{\alpha_r}\alpha_r$$

with the total front and rear cornering stiffness, C_{α_f} and C_{α_r}, respectively. We have to remind here that the assumption in the introduction of this section, that the slip angles are approximately the same for the inner and outer tires on each axle, still applies. We shall also consider this supposition for the total cornering stiffness.

Using the linearization (7.2), we can write for the tire slip angles, α_f and α_r, the following expressions

$$\alpha_f \approx \frac{u_{y,CG} + ar}{u_{x,CG}} - \delta, \qquad \alpha_r \approx \frac{u_{y,CG} - br}{u_{x,CG}}.$$

The sideslip angle at the center of gravity is denoted by β_{CG} (see the figure 7.5). The sideslip angle, β, can be defined at any point on the vehicle body in two ways:

1. If the longitudinal and lateral velocities, u_x and u_y, are given at any point on the vehicle body, we can express the sideslip angle by the following ratio

$$\tan\beta = \frac{u_y}{u_x} \quad \Rightarrow \quad \beta = \tan^{-1}\left(\frac{u_y}{u_x}\right) \approx \frac{u_y}{u_x}.$$

2. The second way how to determine the sideslip angle at any point on the body is as the difference between the direction of velocity, γ, and the vehicle yaw angle (the angle between the longitudinal axis of the vehicle and the fixed axis of the co-ordinate system), ψ, at that point, i.e.

$$\beta = \gamma - \psi.$$

For the determination of the actual vehicle position $[x_0, y_0]$ at time t with regard to the basic system of coordinates we use the following formulas

$$x_0(t) = \int_0^t u_{x_0}\,\mathrm{d}\tau = \int_0^t U_{CG}\cos\left(\beta_{CG} + \psi\right)\mathrm{d}\tau = \int_0^t U_{CG}\cos\gamma\,\mathrm{d}\tau,$$

$$y_0(t) = \int_0^t u_{y_0}\,\mathrm{d}\tau = \int_0^t U_{CG}\sin\left(\beta_{CG} + \psi\right)\mathrm{d}\tau = \int_0^t U_{CG}\sin\gamma\,\mathrm{d}\tau$$

where yaw angle can be obtained from the expression $\dot{\psi} = r$.

Chapter 8

Experiment

For the practical application of the theoretical knowledge in the previous chapters we have to describe the manoeuvre of the test vehicle and also measured signals. The input-output data used for our numerical experiment were obtained in 2001. The experiment was realized by the Institute of Forensic Engineering of Brno University of Technology in conjunction with the Department of Transporting Technology of Brno University of Technology.

The first section deals with the test vehicle, subsequent rubric aims to the test track and the last one describes used measuring equipment. For deeper understanding you can find more information in [1].

8.1 Test Car

The passenger vehicle was utilized for the test manoeuvre. The technical parameters of this vehicle which are necessary for the numerical computation are summarized in the following table.

Notation	**Parameter**	**Measured value**
m	Mass of vehicle	$1446\,\mathrm{kg}$
a	Distance between front axle and center of gravity	$987\,\mathrm{mm}$
a	Distance between rear axle and center of gravity	$1525\,\mathrm{mm}$
i_r	Steering ratio	21
I_z	Yaw moment of inertia	$2319\,\mathrm{kg}\cdot\mathrm{m}^2$
$C_{\alpha f}$	Total front cornering stiffness	$88107{,}7\,\mathrm{N}\cdot\mathrm{rad}^{-1}$
$C_{\alpha r}$	Total rear cornering stiffness	$88107{,}7\,\mathrm{N}\cdot\mathrm{rad}^{-1}$

Table 8.1: The technical specification of test vehicle

8.2 Test Track

The test track was used in accordance with the ISO standard ISO/WD 3888-2 (see [6]) which defines the dimensions of the test track for closed-loop control, specifically determine the severe lane-change manoeuvre test for the obstacle avoidance performance of a vehicle. It is applicable to passenger cars as defined in ISO 3833 and light commercial vehicles up to a gross vehicle mass of 3,5 tonnes. The test track was passed along with an approximately constant velocity.

8.3 Measuring Equipment

Two types of sensors were located on the test car and used for the measuring of some dynamic parameters. More details about the first type of used sensors can be studied in [14].

DATRON-CORRSYS GmbH

- ◇ **HS-CE**, **V1** - vector sensors for a velocity measuring and slip angle measuring,
- ◇ **MSW** - a sensor for a steer angle measuring.

Marking device - an equipment for creating a water lane on the carriageway serving for registering a trajectory of vehicle motion.

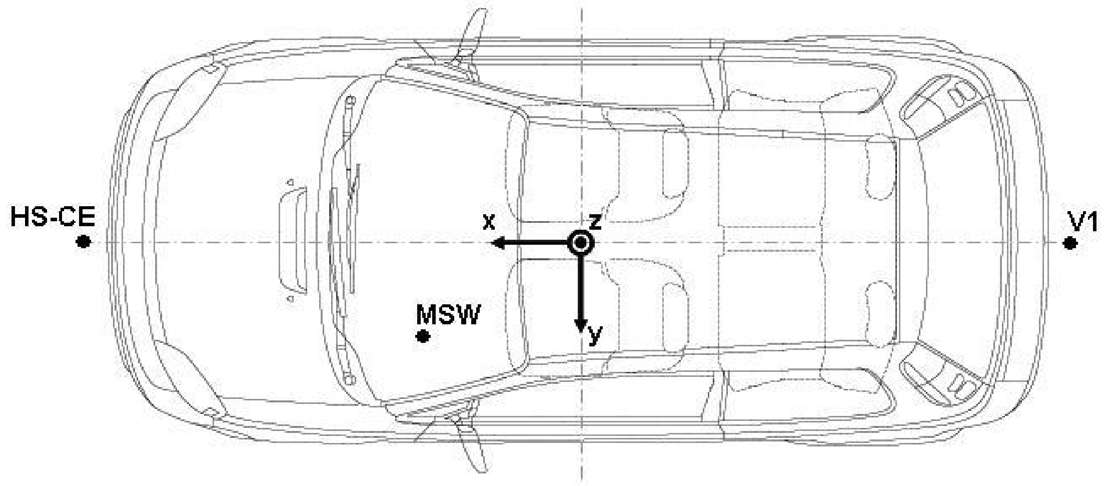

Figure 8.1: Vehicle with measuring devices

A sampling period of all sensors was $hs = 0.1\,\mathrm{s}$ and the manoeuvre lasted $T = 7\,\mathrm{s}$. It appears from this that every sensor recorded seventy measuring values. We can suspect, by reason of low sampling frequency regarding time period T, that the results of the adaptive filter application to this experiment will not be so obvious. We can assume that this effect is caused by the little time period (i.e. by low number of samples) for the adaptation process of the filter.

As we can see on the figure 8.1 vector sensors HS-CE a V1 not lie in the center of gravity. Therefore measured signals do not correspond with the vehicle dynamics in the center of gravity and consequently with the model and equations which we presented in the previous chapter. Thus we have to know the position coordinates of this sensors in the vehicle system of coordinates

$$\begin{aligned}[x_{HS-CE};\, y_{HS-CE},] &= [2,109;\ -0,027]\,,\\ [x_{V1};\, y_{V1},] &= [-2,760;\ -0,379]\,.\end{aligned}$$

Three graphs 8.3, 8.4, and 8.2 illustrate the evolution of the measured signals. From the data obtained by vector sensors HS-CE and V1 may be derived other needful parameters, namely a longitudinal and lateral velocity, u_x and u_y, respectively and yaw rate r. More details of this derivation is possible to find in [12]. The evolutions of derived measured signals are figured in appendix.

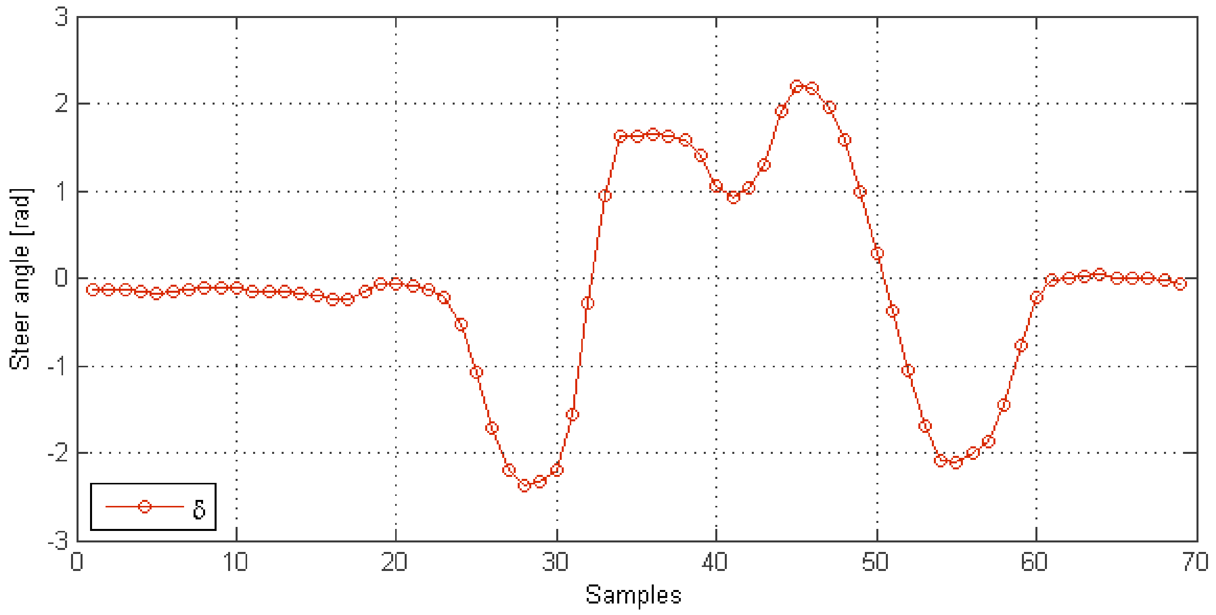

Figure 8.2: Steer angle δ

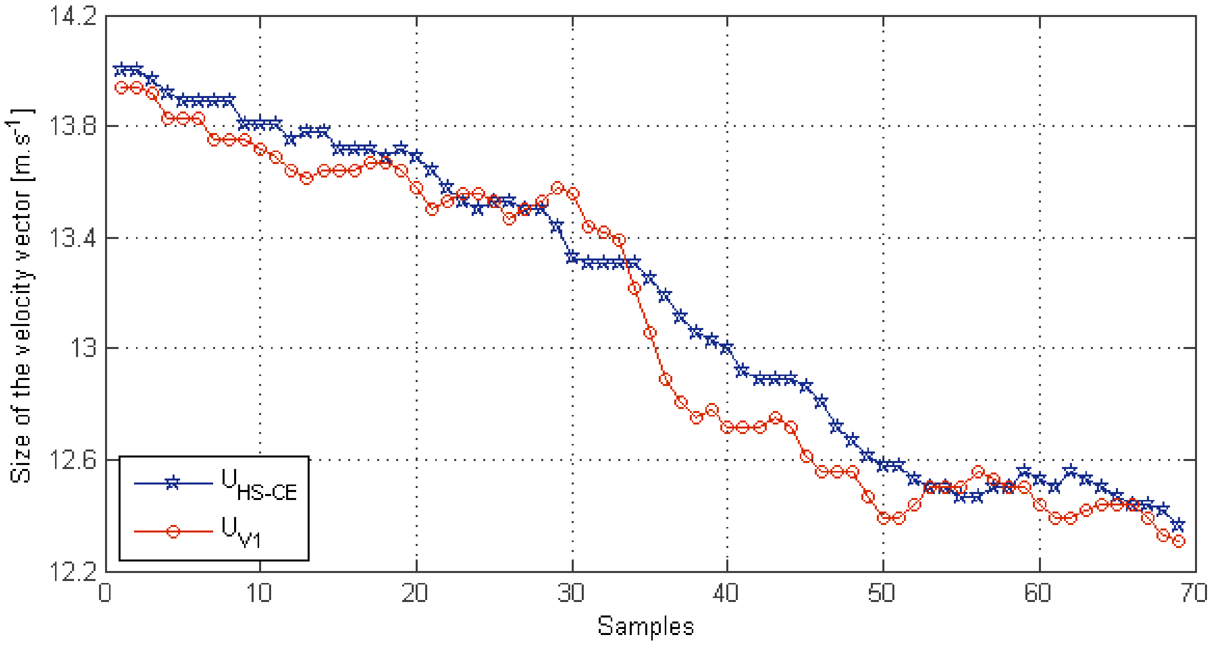

Figure 8.3: Size of the velocity vector U

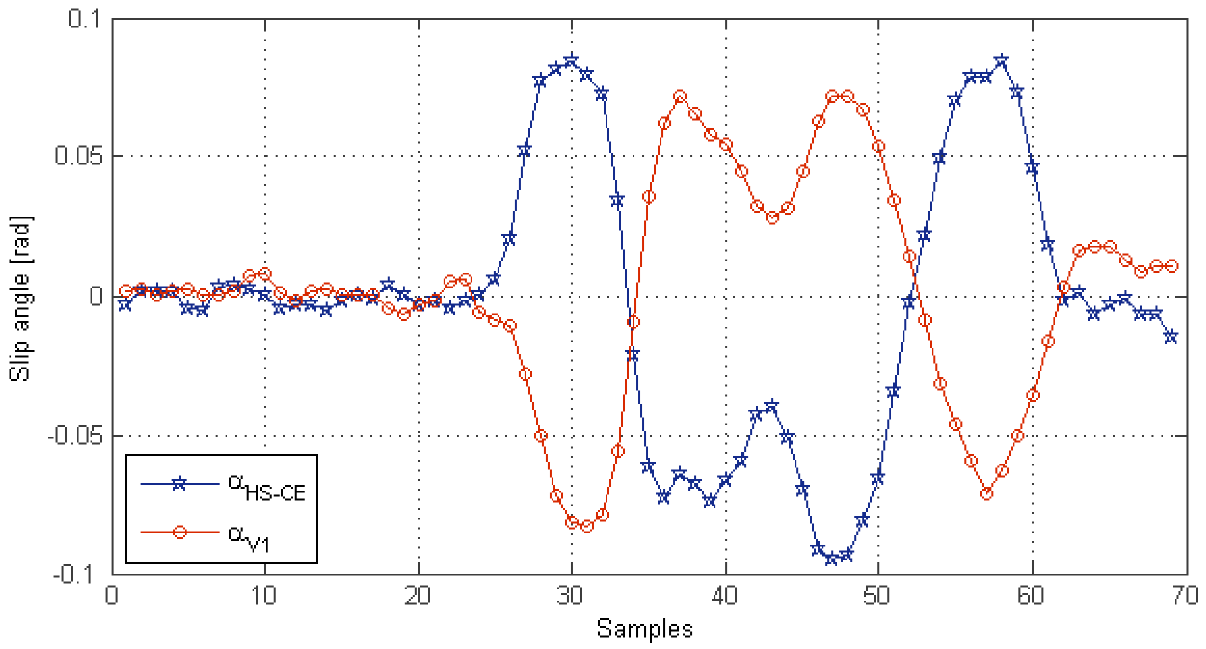

Figure 8.4: Slip angle α

Chapter 9

Application of Recursive Algorithm

In previous chapters we collected all necessary information for the final simulation. Now, we have to define which variables represent the input, state and output of the system and also determine which measured variables will be used for an implementation of the recursive algorithm. The first section will deal with representation of the model and its discretization. Next section will contain two m-files with the recursive algorithm and discretization of state equation. The end of this chapter will cover the results of estimates and their comparison with the measured data.

9.1 Model

In chapter 7 we introduced the single-track model for a description of vehicle lateral dynamics. Forasmuch as we solve application of discrete adaptive filter for vehicle dynamics analysis, it is necessary to transfer this continuous model

$$\begin{bmatrix} \dot{u}_{y,CG} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{-C_{\alpha_f}-C_{\alpha_r}}{mu_{x,CG}} & -u_{x,CG}-\frac{C_{\alpha_f}a-C_{\alpha_r}b}{mu_{x,CG}} \\ \frac{-C_{\alpha_f}a-C_{\alpha_r}b}{I_z u_{x,CG}} & \frac{-C_{\alpha_f}a^2-C_{\alpha_r}b^2}{I_z u_{x,CG}} \end{bmatrix} \begin{bmatrix} u_{y,CG} \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{mi_r} \\ \frac{C_{\alpha_f}a}{I_z i_r} \end{bmatrix} \delta$$

to the discrete one. For this purpose we use the common fourth order Runge-Kutta method which overwrite the system of differential equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t,\boldsymbol{x}) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t)$$

to the system of difference equations by using the following expression

$$\boldsymbol{x}_{i+1} = \boldsymbol{x_i} + \frac{h}{6}(\boldsymbol{k_1} + 2\boldsymbol{k_2} + 2\boldsymbol{k_3} + \boldsymbol{k_4})$$

with

$$\begin{aligned} \boldsymbol{k}_1 &= \boldsymbol{f}(t_i, \boldsymbol{x}_i), & \boldsymbol{k}_3 &= \boldsymbol{f}(t_i + \tfrac{h}{2}, \boldsymbol{x_i} + \tfrac{h}{2}\boldsymbol{k_2}), \\ \boldsymbol{k}_2 &= \boldsymbol{f}(t_i + \tfrac{h}{2}, \boldsymbol{x_i} + \tfrac{h}{2}\boldsymbol{k_1}), & \boldsymbol{k}_4 &= \boldsymbol{f}(t_i + h, \boldsymbol{x_i} + h\boldsymbol{k_3}) \end{aligned}$$

where h is step of numerical method given by formula $h = t_{i+1} - t_i$. In our case we set the step of numerical method $h = 0,01\,\mathrm{s}$. By this discretization process we obtain the following system of difference equations

$$\boldsymbol{x}_{i+1} = \boldsymbol{\Phi}_i \boldsymbol{x}_i + \boldsymbol{\Gamma}_i \boldsymbol{u}_i$$

where $\boldsymbol{\Phi}_i$ is a matrix of dynamic coefficients and $\boldsymbol{\Gamma}_i$ is a matrix interconnecting the inputs with the states. The corresponding measurement equation is expressed by representation

$$\boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{x}_i$$

where matrix $\boldsymbol{H}_i$ interconnecting the output with the state is an identity matrix

$$\boldsymbol{H}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Whence it follows that the output and state vectors are equal.

The original model given by (7.4) then has a following discrete form

$$\begin{bmatrix} u_{y,CG_{i+1}} \\ r_{i+1} \end{bmatrix} = \boldsymbol{\Phi}_i \begin{bmatrix} u_{y,CG_i} \\ r_i \end{bmatrix} + \boldsymbol{\Gamma}_i \delta_i \qquad (9.1)$$

and we can create the table which determine the input, state, output variables of the system.

Input Variable	**State Variables**	**Output Variables**
steer angle δ	lateral velocity $u_{y,CG}$	lateral velocity $u_{y,CG}$
	yaw rate r	yaw rate r

Table 9.1: Mathematical Model for Implementation of Recursive Algorithm

In (9.1) can be found two time-variant variables, namely a longitudinal velocity $u_{x,CG}$, and steer angle δ. Both these variables were measured with the sampling period $hs = 0,1\,\mathrm{s}$. As was mentioned before, the step of the numerical method is $h = 0,01\,\mathrm{s}$ to obtain the more accurately mathematical model. For the inputs of the system measured with period $hs = 0,1\,\mathrm{s}$ we apply an implemented function in MATLAB called '**pchip**' (Piecewise Cubic Hermite Interpolating Polynomial) with syntax $yi = pchip(x, y, xi)$. This function returns vector yi containing elements corresponding to the elements of xi and determined by piecewise cubic interpolation within vectors x and y. The vector x specifies the points at which the data y is given. The time interval $T = 7\,\mathrm{s}$ of the test manoeuvre will thus be divided step-by-step corresponding to the step of numerical method and in these points the values for the system input will be given by this implemented method.

9.2 MATLAB Files

This part includes two m-files with the recursive algorithm, and the discretization of the state equation 7.4. Also it contains figures illustrative an output of our MATLAB program.

9.2.1 M-file for Recursive Algorithm

As the title prompts, this subsection covers an m-file with the recursive least square algorithm used for our simulation. The inputs and outputs applied in function 'rls' are explained in the comments with the corresponding dimensions of variables.

```
function [est,err,theta,evolt] = rls(inpo,outo,inp,y,ddelta,lam,maintitle)

% Inputs:
% inpo,outo = number of inputs and output feedback, dim 1x1,
% inp = input signal, dim Nx1,
% y = desired signal, dim Nx1,
% ddelta = initial value, P(0)=delta^{-1}*I, dim 1x1,
% lam = forgetting factor, dim 1x1, default 0.999.

% Outputs:
% est = estimate, dim Nx1,
% err = estimate error, dim Nx1,
% theta = final value of filter coefficient, dim Mx1,
% evolt = evolution of filter coefficients, dim MxN.

d = inpo + outo;               %filter order
N = size(y,1);
P = ddelta*eye(d);

% Initial weights
theta = zeros(d,1);

for n = inpo :  N
    phi = inp(n:-1:n-inpo+1);     % inp(n),inp(n-1),...,inp(n-inpo+1)
    outp = y(n-1:-1:n-outo);      % y(n-1),y(n-2),...,y(n-outo)
    phi = [phi ; outp];
    g = (phi' * P)'/(lam + phi' * P * phi );
    est(n) = theta' * phi;
    xi(n) = y(n) - est(n);
    theta = theta + g * xi(n);
    P = ( P - g * phi' * P ) / lam;
    evolt(1:d,n) = theta;
end
```

9.2.2 M-file for Discretization of State Equation

The previous section 9.1 was engaged with the discretization of the state equation describing the lateral vehicle dynamics. As was mentioned before, for the dicretization of model was chosen the fourth order Runge-Kutta method. Here, we shall state source code containing this dicretization method for the state equation. Again, in this code the information about the inputs of the function 'stateEq' is given in the comment environment. This function makes use of another function of input, state, and noise variable called 'bodydot' which is in fact multiplied state equation (7.4).

```
function x1 = stateEq(x,u,v, h)     %x state, u input, v noise, h step

%Discretization
k1 = h*bodydot(x,u,v);
k2 = h*bodydot(x+(h/2)*k1,u,v);
k3 = h*bodydot(x+(h/2)*k2,u,v);
k4 = h*bodydot(x+h*k3,u,v);
x1 = x + (k1+2*k2+2*k3+k4)/6;

function xdot = bodydot(x,u,v)

m = 1446;
Caf = 88107.7;
Car = 88107.7;
Iz = 2319;
a = 0.987;
b = 1.525;
ir = 21;

% x = [Uy r] u = [Delta Ux]
xdot=[-x(1)*((Caf+Car)/(u(2)*m)) + x(2)*((-Caf*a+Car*b-u(2)^2*m)/(u(2)*m))
    + u(1)*Caf/(m*ir) + v(1);
    x(1)*((-Caf*a-Car*b)/(u(2)*Iz)) + x(2)*((-Caf*a^2-Car*b^2)/(u(2)*Iz))
    + Caf*a*u(1)/(ir*Iz) + v(2)];
```

Both these functions, namely 'rls' and 'stateEq', are applied in the m-file entitled experiment.m. Hence, we can consider this file as a principle of our MATLAB program whose output is represented in the following figure 9.1. This figure contains two windows which appear by calling command experiment. The first one shows an estimate of the system output without usage of the function 'pchip' and the second one, on the contrary, with the application of implemented MATLAB function 'pchip'. They figure the evolutions of the system output, error signal, and filter coefficients.

Remark. The figure 9.1 illustrates the output of our program where an estimate pertains to the first component of the state vector, i.e. the lateral velocity.

```
>> experiment
```

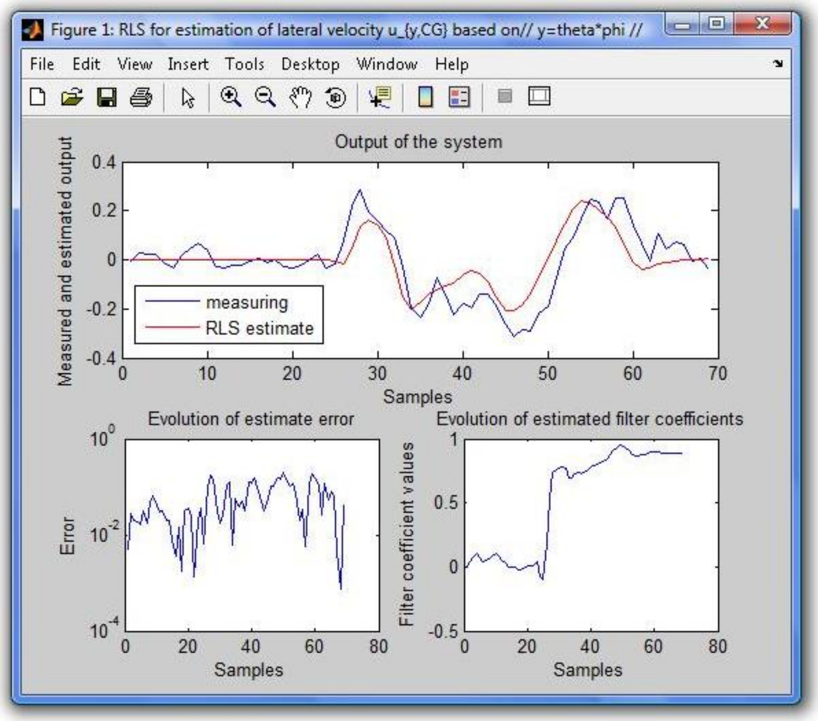
Figure 1: RLS for estimation of lateral velocity u_{y,CG} based on// y=theta*phi //
File Edit View Insert Tools Desktop Window Help
Output of the system
Measured and estimated output
measuring
RLS estimate
Samples
Evolution of estimate error
Error
Samples
Evolution of estimated filter coefficients
Filter coefficient values
Samples

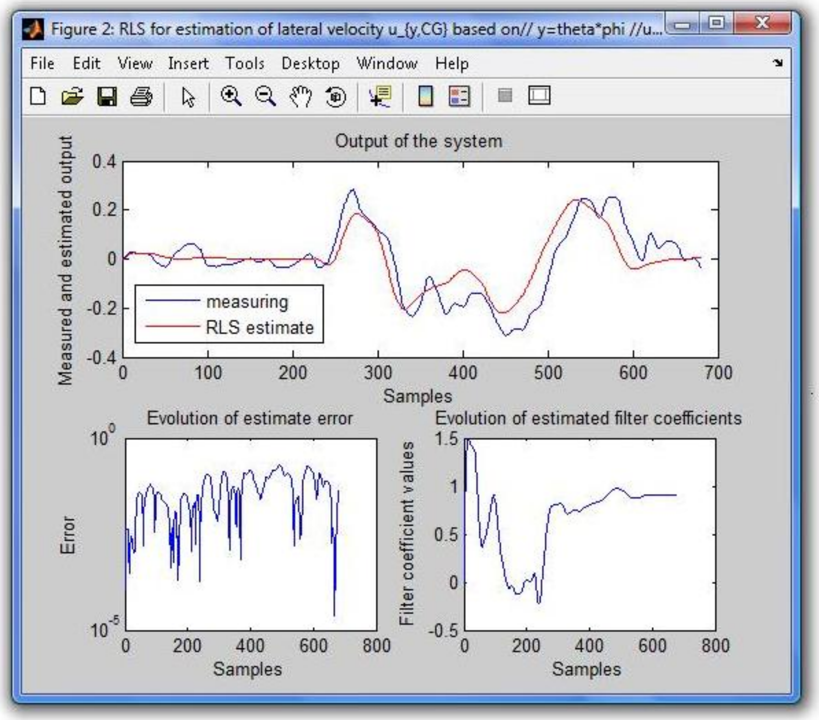
Figure 2: RLS for estimation of lateral velocity u_{y,CG} based on// y=theta*phi //u...
File Edit View Insert Tools Desktop Window Help
Output of the system
Measured and estimated output
measuring
RLS estimate
Samples
Evolution of estimate error
Error
Samples
Evolution of estimated filter coefficients
Filter coefficient values
Samples

Figure 9.1: Output of MATLAB program

9.3 Results

In this section, we will discuss the results obtained by application of the recursive least squares algorithm to the experimental data. We will compare filtered out state variables with their measured values and also show in the graphs the evolution of corresponding filter coefficients. We will use only the output of our MATLAB program, which does not employ the 'pchip' function.

9.3.1 Results for State Variables

The first two figures are related to the lateral velocity and the second two to the yaw rate. In either event, the introductory figure from this pair illustrates the comparing of measured and filtered out values of state variable and the following graph represents the evolution of the corresponding filter coefficients.

In our simulations we used also another implemented MATLAB function called '**awgn**' (Add White Gaussian Noise) with the syntax $y = awgn(x, snr,' measured')$. This function adds white Gaussian noise to the vector signal x and measures the power of x before adding noise. The scalar snr specifies the signal-to-noise ratio per sample, in dB. For this reason the output of our program (and of course the estimated output of the system) is erratic.

We also had to set the values of parameters occurring in the recursive algorithm. For its initialization δ is only the required parameter. Further, we chose the constant value for the forgetting factor because the system changes gradually and does not undergo abrupt variances. We elected following values for this two parameters

$$\delta = 10^2, \qquad \lambda = 0,999.$$

In the picture 9.2, we can see that the filtered out values of the lateral velocity do not copy completely the curve of the measured values. Nevertheless the estimated lateral velocity catches approximately the evolution of the measured state variable, though the filter does not react on the small velocity changes. Similar effect can be remarked also in the figure 9.4. Here it seems, that the evolution of filtered out values of yaw rate is shifted (or delayed) from around the thirty-fifth sample about two to three samples. This feature brightly matches the curve illustrating the evolvement of corresponding filter coefficients in the figure 9.5. The values of filter coefficients from the mentioned moment are located above one around the value 1,3. We should point out that from the substance of the recursive filter (see the motivation and especially the formula (1.1)) allows that the filter coefficients should move about the value one in the ideal case. This phenomenon can be seen in the figures 9.3 and 9.5 from the number of somewhere thirty samples. But, the values of this parameters are not as close to one as we wish. That is why the estimated values do not duplicate precisely the measured values of state variables.

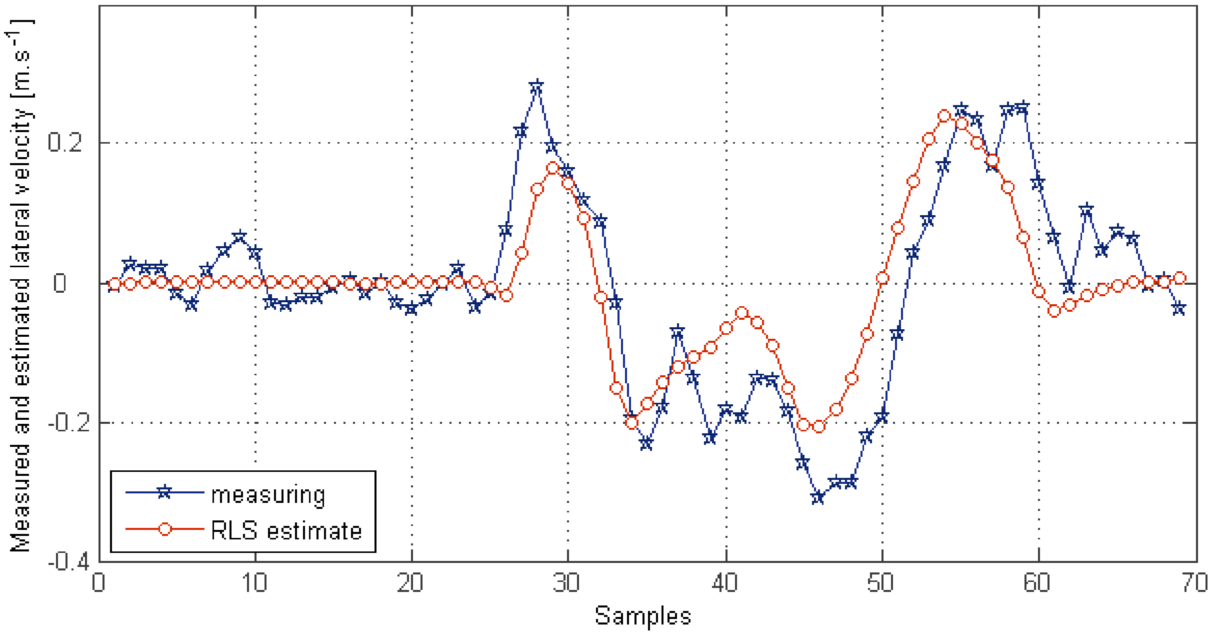

Figure 9.2: Comparing of measured and filtered out values of lateral velocity

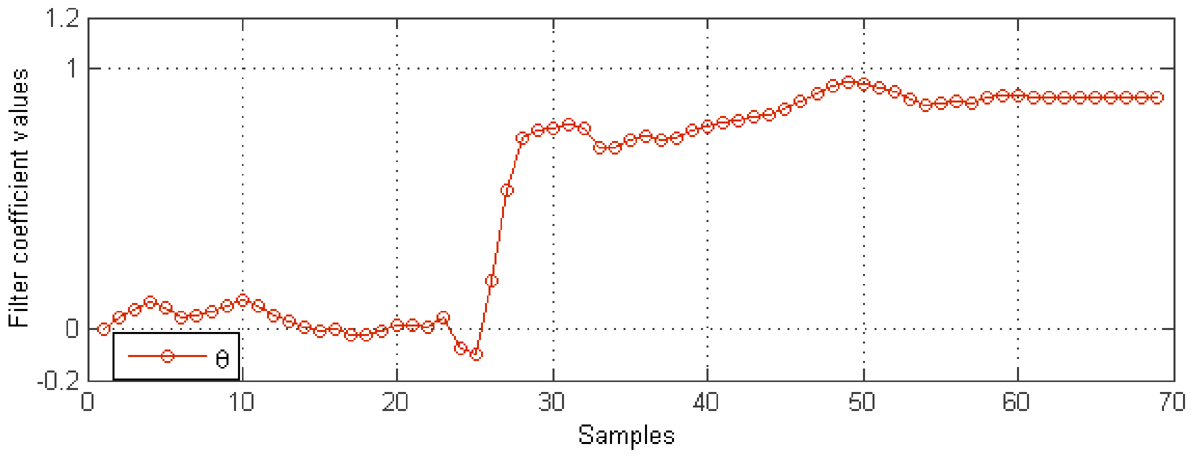

Figure 9.3: Evolution of the filter coefficients for the recursive least squares estimation of lateral velocity

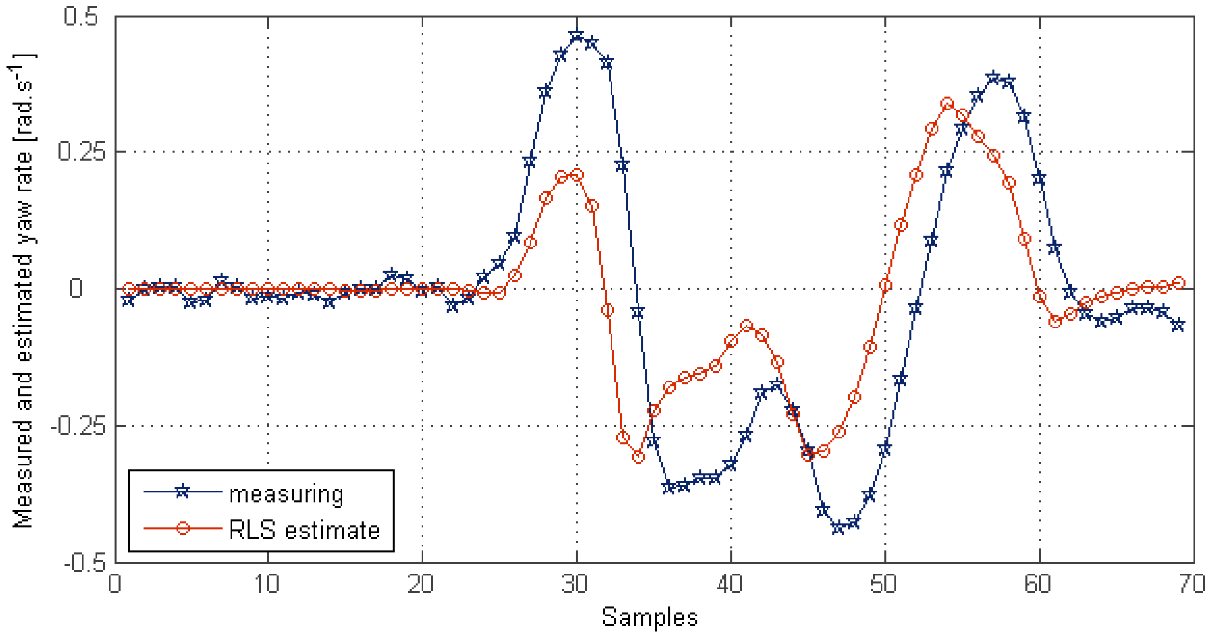

Figure 9.4: Comparing of measured and filtered out values of yaw rate

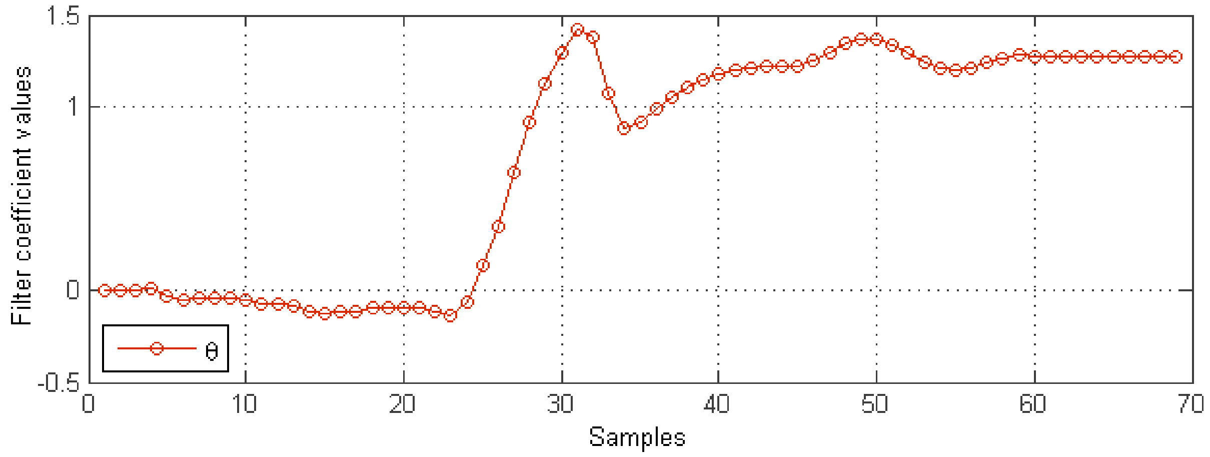

Figure 9.5: Evolution of the filter coefficients for the recursive least squares estimation of yaw rate

9.3.2 Derived Vehicle Position from Estimated State Variables

For the determination of the actual vehicle position we need to know the following velocity components in the basic coordinate system

$$\begin{bmatrix} u_{x0} \\ u_{y0} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{9.2}$$

where the yaw angle ψ can be obtained by integration of the yaw rate. By integration of the velocity components u_x, u_y we get the position data in the basic system of coordinates. Forward Euler was used for these numerical integrations.

The figure 9.6 represents comparison of vehicle position measured by marking device, computed from estimated state variables and rotated derived position. As we can see, the trajectory derived from filtered out state variables (the green curve in 9.6) is slightly rotated, which can be caused by shoulder of sensors on the vehicle. The coordination system of the sensor could be small angled with respect to the coordinate system of the vehicle and as the result was the displacement of the derived trajectory. The discordant direction of vehicle raid on the test track could be another reason. In this case the coordinate system of the vehicle is angled regarding the basic system of coordinates. The red curve in graph 9.6 is the derived trajectory from estimated variables, which is angled by $1,9°$.

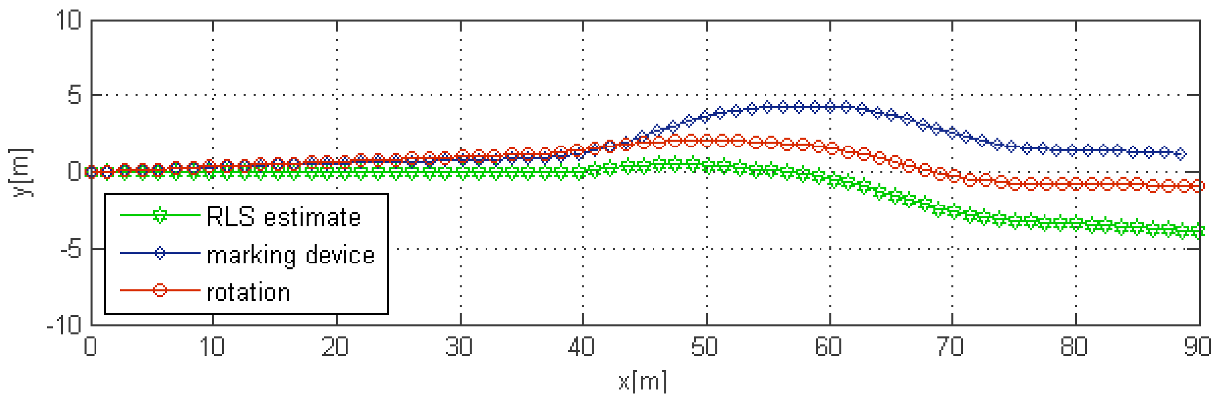

Figure 9.6: Comparing of the position given by marking device, derived from the state variables, and by angular rotation of derived position

9.4 Comparison with Kalman Filter Results

This section compares two algorithms which are based on different estimation methods. As we know, the least squares method is the deterministic method which minimizes the squares of the error signal (the difference between the desired and actual signal) and is fundamental for the recursive algorithm. On the contrary, the Kalman filter algorithm is the stochastic method, which is founded on the minimizing of the mean squared error (least mean squares method). More information about Kalman filter can be looked up in the source [7]. Following figures serve to trade both these algorithm off.

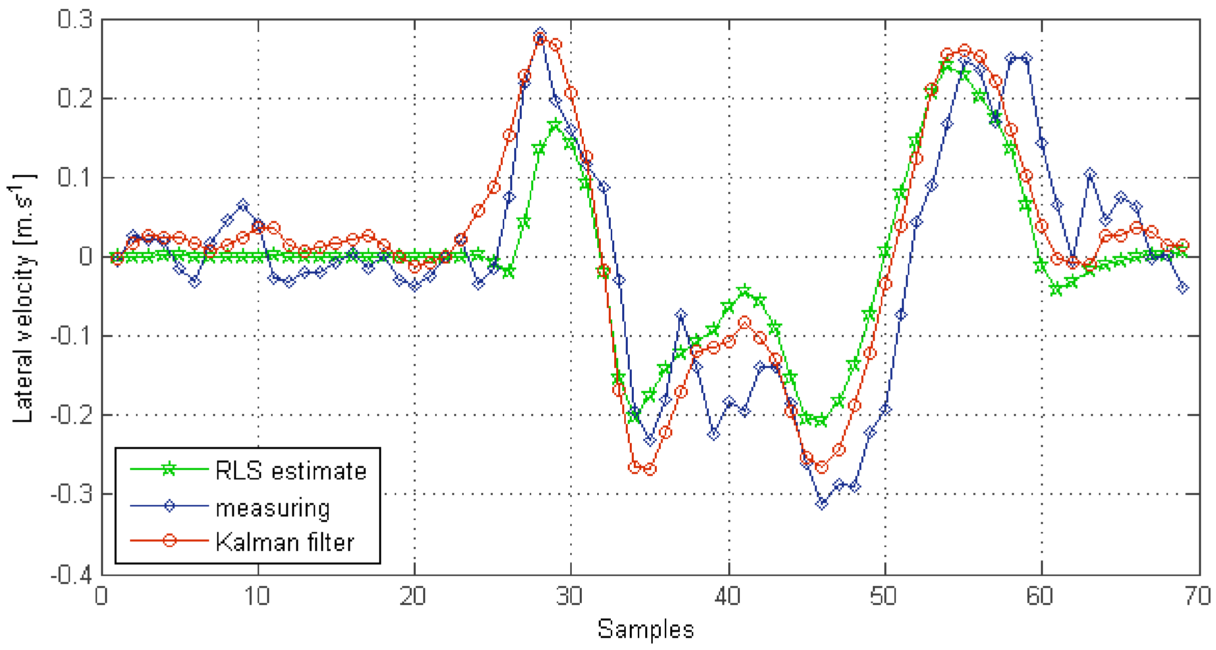

Figure 9.7: Comparing of measured lateral velocity with estimated by the recursive algorithm and by Kalman filter algorithm

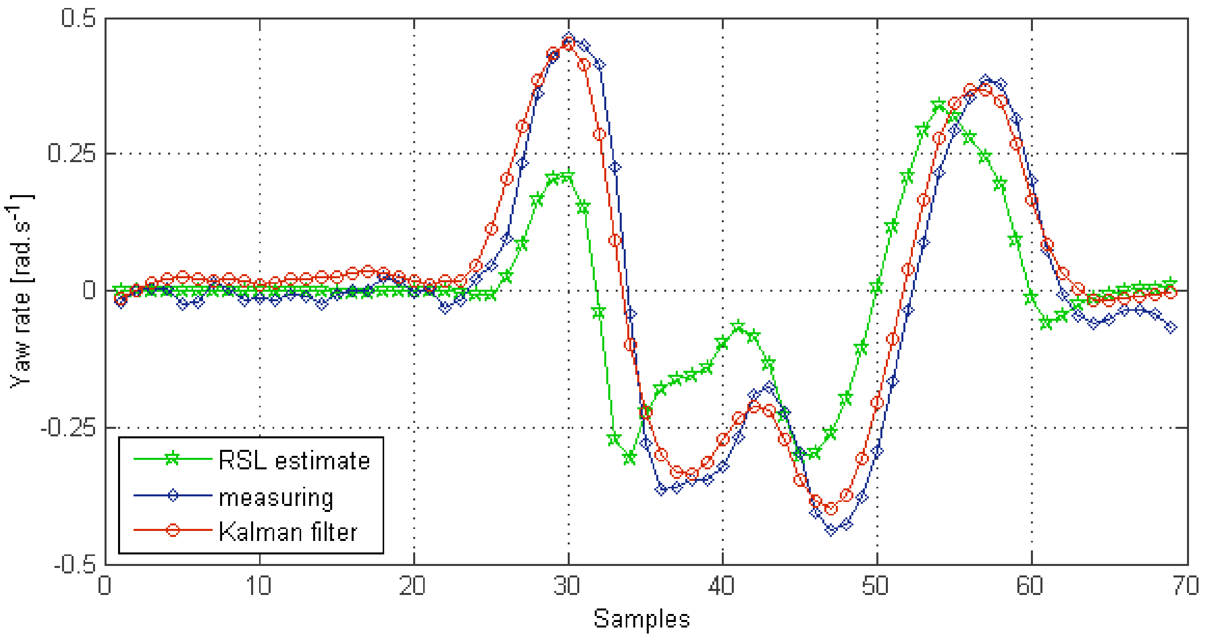

Figure 9.8: Comparing of measured yaw rate with estimated by the recursive algorithm and by Kalman filter algorithm

In the figures 9.7 and 9.8 it is apparent that Kalman filter is more sensitive to the small changes in the evolution of state variables and copies almost exactly the curve of measuring. Thus we can say that Kalman filter is preferable to our recursive algorithm to these experimental data.

The comparison of computed vehicle position from the states estimated by the recursive algorithm and by Kalman filter algorithm is represented by the last figure in this chapter. The derived trajectories angled by $1,9^\circ$ are used for this confrontation. The trajectory measured by marking device is placed between the estimated curves. The red colored trajectory illustrating the vehicle position derived by Kalman filter algorithm renders better the measured vehicle position which arises from the last conclusion that the Kalman filter algorithm is much better estimation method than the recursive least squares algorithm.

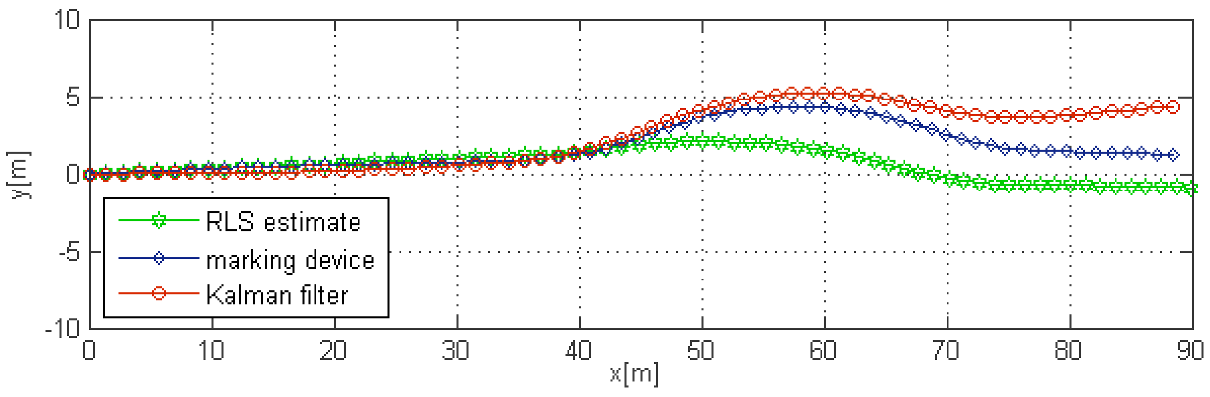

Figure 9.9: Comparing of measured trajectory with derived from state variables estimated by the recursive algorithm and by Kalman filter algorithm

Chapter 10

Conclusion

The goal of this thesis was to initiate readers into the theory of adaptive filters and their application in the automotive practice. For this purpose the mathematical description of dynamic systems was mentioned in the introductory chapters of this work. The models of dynamic systems form the basis of the description of the stochastic systems, whose states we estimated with the help of the adaptive filter relied on the recursive least squares algorithm. So, we focused on the optimal filtering of noisy data in the stochastic systems analysis. The weighted least squares method, stated in the chapter titled Parameter Estimation Methods, then was fundamental for defining of the recursive least squares method and for deriving of the recursive algorithm.

This algorithm with the pieces of knowledge about its initialization and choice of forgetting factor were consequently employed in software solution in the numerical computing environment MATLAB.

The utilization of derived recursive algorithm equations in the automotive practice was mainly aimed at the test car driving manoeuvre analysis. For the description of lateral vehicle dynamics we chose the planar single-track model vehicle and investigated the steering response at the constant driving velocity (steerability). We applied the derived algorithm to the experimental data measured during the lane-change manoeuvre test realized in 2001.

At the close of this thesis the results of the recursive algorithm application were compared with results obtained by Kalman filter application. From the evolutions of the estimated state variables we could come to a conclusion that the Kalman filter is preferable estimator to the recursive algorithm and is more sensitive to small changes of measured signals.

A realization of the recursive least squares algorithm in the interactive environment MATLAB and its application to the vehicle test manoeuvre, even if the description of vehicle dynamics was considerably simplified, are the good principle of the applications for the test manoeuvres exploiting more complicated models.

Appendix A

Evolutions of Derived Measured Signals

This appendix states the completive graphs with the evolutions of derived measured signals, namely longitudinal and lateral velocity and yaw rate. As was mentioned in the chapter with the title Experiment the details about derivation are described in [12]. Wittingly, the curve colors are differentiated accordance with type of variable. In the concrete, the curves illustrating the evolutions of the input variables are colored in red and evolutions of the measured state variables are represented by the blue color.

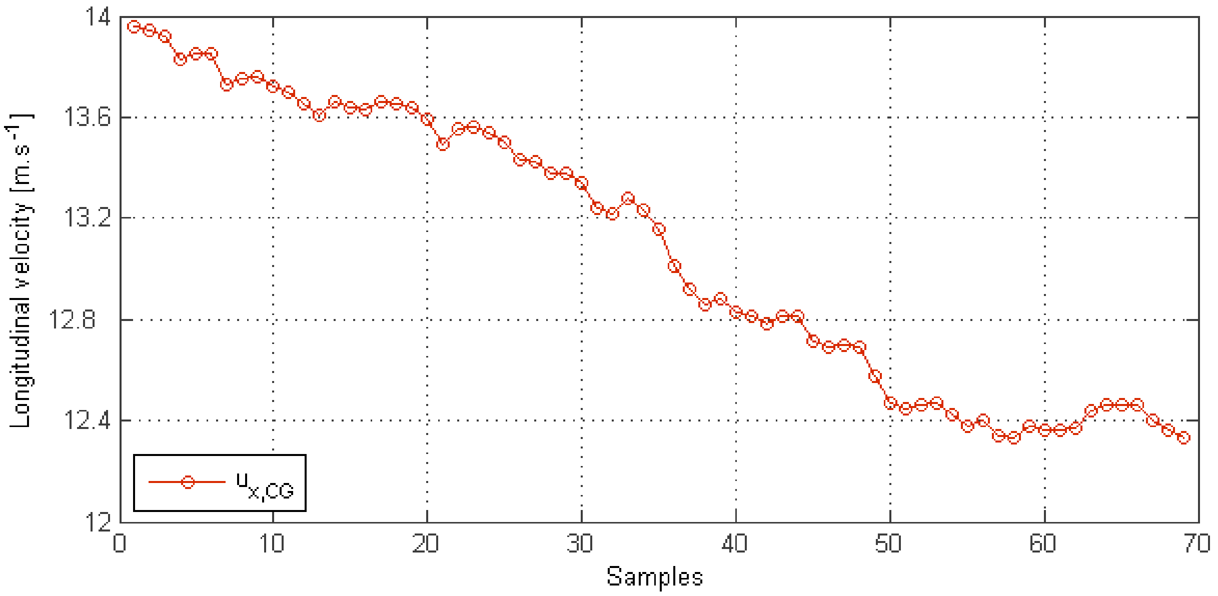

Figure A.1: Longitudinal velocity u_x

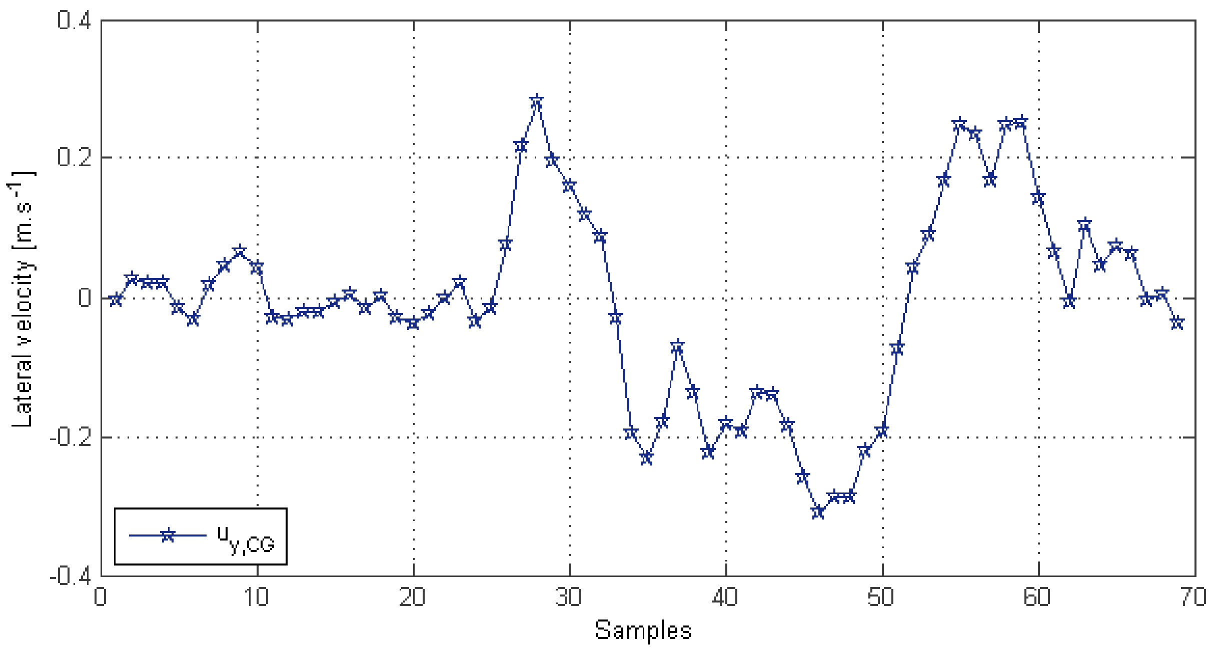

Figure A.2: Lateral velocity u_y

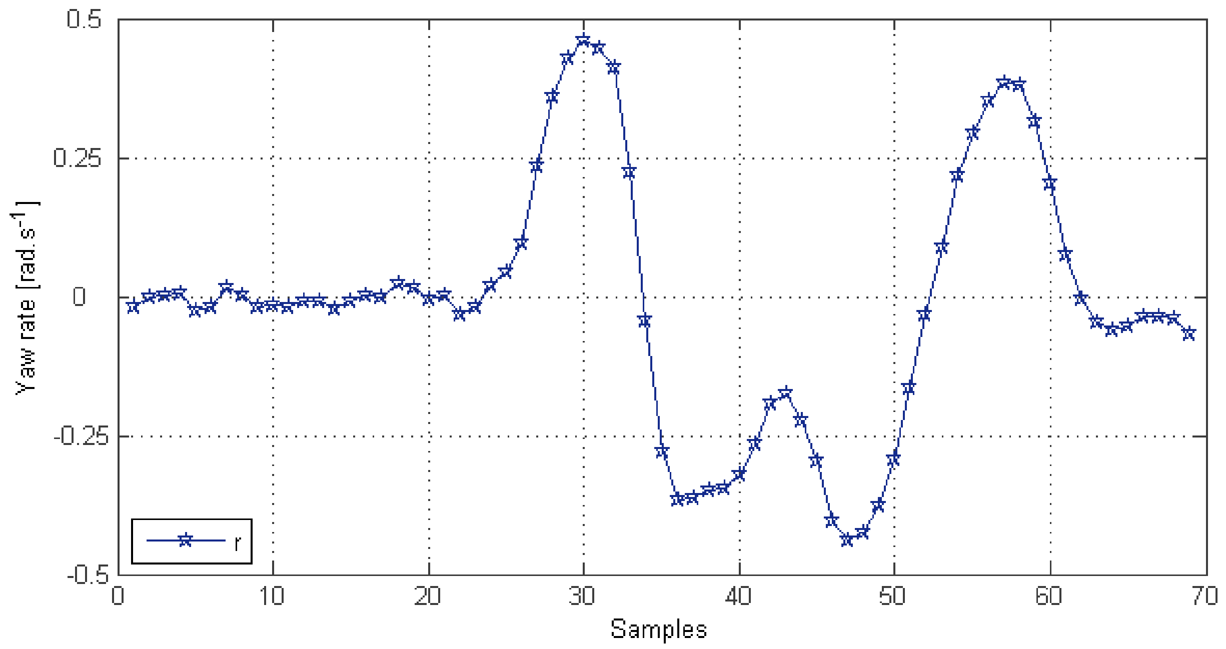

Figure A.3: Yaw rate r

Bibliography

[1] BLAŤÁK, Ondřej.: *Analýza jízdního manévru vozidla*. Brno, 2003. Diploma thesis. Vysoké učení technické, Fakulta strojního inženýrství.

[2] EVANS, Lawrence Craig.: *An Introduction to stochastic differential equations*. Version 1.2. Department of Mathematics UC Berkeley. 139 pgs.

[3] HAVLENA, Vladimír; ŠTECHA, Jan.: *Moderní teorie řízení*. Praha: Skriptum ČVUT, FEL, 2000. 297 pgs. ISBN 80-01-02095-9.

[4] HAYES, Monson.: *Statistical Digital Signal Processing and Modeling*. 1st edition. Georgia Institute of Technology: JOHN WILEY & SONS INC., 1996. 608 pgs. ISBN 0-47159431-8.

[5] HAYKIN, Simon.: *Adaptive Filter Theory*. 3rd edition. Berkley: Prentice Hall Information and system science series, 1996. 920 pgs. ISBN 978-0-130-90126-2.

[6] ISO WD 3888-2. *Passenger cars – Test track for a severe lane change manoeuvre – – Part 2: Obstacle avoidance*. 2002. 6 pgs.

[7] LAURINEC, Marián.: *Využití Kalmanova filtru pro analýzu dynamiky vozidel*. Brno, 2007. 67 pgs. Diploma thesis. Vysoké učení technické, Fakulta strojního inženýrství.

[8] LJUNG, Lennart.: *System Identification: Theory for the Users*. 2nd edition. New Jersey: Prentice Hall PTR, 1999. 609 pgs. ISBN 0-13-656695-2.

[9] KIENCKE, Uwe; NIELSEN, Lars.: *Automotive Control Systems: For Engine, Driveline, and Vehicle*. 2nd edition. Berlin: Springer, 2005. 512 pgs. ISBN 978-3-540-23139-4.

[10] PUGACHEV, Vladimir Semenovich; SINITSYN, Igor Nikolaevich.: *Stochastic Systems: Theory and Applications*. 3rd edition. London: World Scientific, 2001. 908 pgs. ISBN 978-9-810-24742-3.

[11] RYU, Jihan.: *State and Parameter Estimation for Vehicle Dynamics Control Using GPS*. 2004. 125 pgs. Dissertation. Stanford University

[12] VLK, František.: *Dynamika motorových vozidel*. 2nd edition. Brno: Prof. Ing. František Vlk, DrSc., nakladatelství a vydavatelství, 2003. 432 pgs. ISBN 80-239-0024-2.

[13] Bakal06.chytrak.cz: Státnice ČVUT-FEL 2006 Bc.: *Dynamické systémy.* [online]. URL: <http://www.bakal06.chytrak.cz/2006_K37.pdf>. [cited December 10, 2009]

[14] *Complete Vehicle Testing Solutions: Sensors and Accessories.* [online]. 2008. last revision: September 23, 2009. CORRSYS-DATRON: GPS Sensors - MicroSAT. URL: <http:www.corrsys-datron.comgps_sensors.htm>. [cited April 3, 2010].

[15] RIEVELEY, Rob.: *Evaluation of Numerical Vehicle Dynamics Simulation Techniques.* [online]. University of Windsor, 2006. Graduate Seminar Presentation.
URL: <http://137.207.14.230/ ~web/vdc/downloads/library/presentations/Grad_ _Sem_31JA06_RR.pdf>. [cited January 19, 2010]

The List of Used Shortcuts and Symbols

CG	Center of Gravity
LLS	Linear Least Squares
NLS	Nonlinear Least Squares
RLS	Recursive Least Squares
WLS	Weighted Least Squares

$\boldsymbol{u}$	system input
$\boldsymbol{x}$	system state
$\boldsymbol{y}$	system output
$\boldsymbol{\varphi}$	vector of regressors
$\boldsymbol{\theta}$	vector of filter coefficients
d	number of filter coefficients (dimension of the filter coefficients vector)
$\hat{y}(t\|\boldsymbol{\theta})$	predictor
$\boldsymbol{F}$	matrix of system
$\boldsymbol{G}, \boldsymbol{\Gamma}$	matrix interconnecting the input with the state of the system
$\boldsymbol{H}$	matrix of measuring sensitivity
$\boldsymbol{C}$	matrix interconnecting the input with the output of the system
$\boldsymbol{\Phi}$	fundamental matrix
$\boldsymbol{\Phi}(t, \tau)$	state transition matrix
$\boldsymbol{I}$	identity matrix

$\mathcal{L}$	Laplace operator
$\mathrm{E}(X), \mu_X$	expectation value of random variable X
$\mathrm{Var}(X), \sigma_X^2$	variance of random variable X
$\mathrm{Cov}(X,Y), c_{XY}$	covariance of random variables X and Y
$c_{XY}(t_1,t_2)$	cross-covariance of random variables X and Y
$\boldsymbol{C_X}$	autocovariance matrix of random process X
r_{XY}	correlation of random variables X and Y
$r_{XY}(t_1,t_2)$	cross-correlation of random variables X and Y
ρ_{XY}	correlation coefficient of random variables X and Y
$\boldsymbol{R_X}$	autocorrelation matrix of random process X
$\mathcal{B}$	bias
δ	Dirac delta in chapter 4, constant for initialization in chapter 6, steering angle in chapter 7
Δ	forward difference in chapter 3, Kronecker delta in chapter 4
$\boldsymbol{v}$	noise of process
$\boldsymbol{w}$	noise of measurement
$\boldsymbol{b}$	Brownian motion (Wiener process)
$\mathcal{M}$	model structure
$\mathcal{M}^\star$	set of models
Z^N	batch of data of the inputs and outputs over the time interval $1 \leq t \leq N$
$\hat{\theta}$	estimate of filter parameters
$V_N(\theta, Z^N)$	criterion function
ε	predictor error (a posteriori error)
ξ	a priori error
β	weight in chapter 5, sideslip angle in chapter 7
λ	forgetting factor
$\boldsymbol{P}$	inverse of autocorrelation matrix $\boldsymbol{R}$

$\boldsymbol{g}$	gain vector
F_y	lateral tire force
α	slip angle
C_α	cornering stiffness
$F_{y,f}$	front lateral tire force
$F_{y,r}$	rear lateral tire force
α_f	front slip angle
α_r	rear slip angle
u_x	longitudinal velocity
u_y	lateral velocity
I_y	yaw moment of inertia
m	vehicle mass
a	distance of the front axle from the center of gravity
b	distance of the rear axle from the center of gravity
a_y	lateral vehicle acceleration
r	yaw rate
ψ	yaw angle
C_{α_f}	total front cornering stiffness
C_{α_r}	total rear cornering stiffness
γ	direction of velocity
x_0, y_0	data position in basic system of coordinates
U	size of the velocity vector in the center of gravity
i_r	steering ratio
hs	sampling period
h	step of Runge-Kutta method