Óbuda University

PhD thesis

New adaptive methods for Robust Fixed Point Transformations-based control of nonlinear systems

by

Teréz A. Várkonyi

Supervisor:

Dr. József K. Tar
Doctoral School of Applied Informatics, Óbuda University

Budapest, 2013

Members of the comprehensive exam committee:

Members of the defense committee:

1. Reviewer:

2. Reviewer:

Day of the defense: 05/24/2013

Signature from head of PhD committee:

To my parents who led me through all those years - it must have been a tough job.

And to Fazekas Mihály Primary and Secondary Grammar School - well, even black sheep can discolor in time.

Acknowledgements

I would like to thank God for his permanent support and that He made all these things possible.

I would like to acknowledge the tremendous help and support of my supervisor, Professor József K. Tar. I also would like to thank the guidance and vast help of Professor Vincenzo Piuri.

I would like to thank Professor Imre J. Rudas his great support and encourage for writing this thesis and that he ensured my working conditions.

I would like to thank the leaders of the Doctoral School of Applied Informatics at Óbuda University, Professor Aurél Galántai and Professor László Horváth, and that of the Doctoral School of Computer Science at Universitá degli Studi di Milano, Professor Ernesto Damiani, for giving me the opportunity to take my first steps on the research career.

I also would like to acknowledge the base, the support, and the vast help in the review of my mother, Professor Annamária R. Várkonyi-Kóczy.

I am very grateful of Óbuda University and Universitá degli Studi di Milano for the support and the arrangement of the double degree program.

I would like to thank my teachers, Professor József Dombi, Professor András Rövid, Professor Márta Takács, and Professor János Fodor, for giving me insight to new scientific areas during these years.

I am very grateful of Professor Csaba Szabó for his patience and trust.

Special thanks to Ruggero Donida Labati for the template of the thesis.

Last but not least, I would like to thank my family without whom I could not have written this thesis.

Contents

1

Introduction

Nowadays, system control is essential in everyday life. It has a long history, e.g. it was applied already by the Romans to handle irrigation systems. In our days, the machines, like mechanical and electronic systems (from the excavators to the CD players) are unimaginable without control.

In the present, as a part of control, one of the most prevalent topics is the control of systems with uncertainties. The growing expectations of avoiding human assistance in situations that need increased attention because of the system's vagueness or dangerousness makes the role of the automated controllers (that can handle vagueness) increased. Just to mention some examples, the automated control of power plants [1], trains [2], or the now-tested "artificial drivers" for cars [3] are like this.

The uncertainties of systems can be divided into three main groups: 1. when the system contains unknown parameters 2. when the system has unknown dynamics 3. when the the system's state cannot be measured [4]. There are many possible ways how to control such systems, e.g. using as much a priori knowledge as possible, using the linear parametrization method, and/or applying learning mechanisms to gain more information about the uncertainty. After that many controllers can be designed for the system, for example sliding mode controllers [5, 6, 7], fuzzy logic controllers [8, 9], anytime controllers [10, 11], neural network controllers [12, 13, 14], fault tolerant controllers [15, 16], and robust controllers [17, 18]. When the system is not linear in its parameters different adaptive controllers can be designed, like [19].

When the controlled system is just partly known robust controllers bring the most benefit. They have been designed and investigated since the 1950s [20]. Since the first applications the area has started a fast progress because the first methods have been sometimes found to lack robustness. The other problem was that in some cases when Sliding Mode Controller, one of first robust controllers [6], was used the actuators have had to cope with high frequency chatter-like control actions that damaged the system. The third reason of the progress was that scientists have realized that robust controllers were very effective when model approximations and disturbances had to be handled in the control process. So the field began to develop. Because of today's higher expectations the topic is still growing.

One of the recent robust control strategies is the method called Robust Fixed Point Transformations (RFPT). It was first designed to overcome the complexity of Lyapunov function-based techniques for smooth systems [21] but after its robustness was improved [22, 23] it became a powerful technique to reduce the disadvantages of the model approximations and disturbances. The method applies the concept of the so-called expected – realized system response and can be used in the environment of traditional feedback and Model Reference Adaptive control systems [24]. In the first applications it was applied only for single input – single output systems but later it was extended to multiple input – multiple output systems, too [25]. Its aim is to make controllers robust in that case when an approximate model is used to estimate the behavior of the system in the control process. Its great advantage is that it can significantly reduce the errors caused by the model approximation and that the disturbances barely affect its performance.

This thesis focuses on improving RFPT, because though it gives the opportunity to avoid the complexity of Lyapunov's method, and it can reduce the disadvantages of the model approximation, there are several questions left open and also disadvantages to get rid of because they make uncertain or even limit the usage. The first drawback among them is that RFPT uses the local attraction of a fixed point. The local attraction means that it gains only local stability according to Lyapunov's stability theorem. This raises the issue if it was possible to achieve its stability.

The other property of RFPT is that theoretically it can improve any existing controller's results if the control task and the controller meet several conditions. Although, up to this point this statement was proved only for classical controllers. So the question if it could ameliorate other types of controllers is open-ended.

The third aspect which is not to be sneezed at is the applicational possibility in real life. On the one hand, there are fields of application that significantly contribute to the improvement of control science. The question is if RFPT could be utilized in these areas. On the other hand, assume that there exists a system or phenomenon which is too complex or some lacking resources (time, knowledge, etc.) do not let it to be modeled accurately. In this case, only a rough approximation of the system can be captured by a model. The main question here is if there is any connection between the analysis of the approximate model and that of the actual system. Is it possible to construct a controller which according to the approximate model's results can properly control the real system? Will the system reach the desired state accurately enough? And finally, since Robust Fixed Point Transformations is specialized in approximate models it is possible that it can improve the accuracy of the above mentioned controller so that the model generates truthful output?

The thesis deals with the above questions and gives positive answers to some of them.

The contributions of this thesis can be summarized as follows:

First of all, a new possible application field for Robust Fixed Point Transformations is investigated. Different chaotic attractors are examined and approximate models are built for them. Then RFPT-based controllers are designed for synchronizing two same type attractors based on the built approximate models. The results show that RFPT is appropriate for chaos synchronization because of its robustness: the performance of the original controllers is significantly increased with RFPT, and a well set controller cannot exceed a poorly adjusted controller with RFPT extension.

Then the mathematical background of Robust Fixed Point Transformations is analyzed. A new structure for RFPT is proposed in which two controllers are integrated to the system. Then it is shown by illustrative examples that the new structure gains an additional tracking error reduction compared to that of the original methods.

After that, the stability of Robust Fixed Point Transformations is considered. An innovative fuzzy-like parameter tuning method is introduced. It is shown that more stable results of RFPT can be gained if the parameter tuning is applied in the control process.

In the sequel the stability of Robust Fixed Point Transformations is reconsidered. A new VS-type stabilization algorithm for RFPT is introduced. The results show that when the RFPT-based controller falls out from the local convergence interval it becomes unstable and generates the so-called chattering effect. In the next step it is shown that the proposed algorithm can reduce the order of chattering and stops it in very short time. As a consequence the stability of the RFPT-based controllers is gained.

Next the combinability of Robust Fixed Point Transformations is studied. Two types of soft computing (SC) based controllers (Fuzzy Logic Controller and Neural Network Controller) are combined with RFPT then compared to their original form. The results verify that the robustness of the controllers can be increased with the application of RFPT and by this the error produced by the original soft-computing-based controllers can be reduced significantly.

Afterwards, the applicability of Robust Fixed Point Transformations is investigated. A hydrodynamic model of freeway traffic is studied from the viewpoint of stability. The stationary solutions of the model are determined and their stability is analyzed. Then an RFPT-based controller is designed to control the emission rate of exhaust fumes for the stationary solutions. Finally, the effectiveness of the controller is examined by comparing its results to the same controller without RFPT.

Finally, preliminary investigations are made for a possible anti-lock braking system. A simple approximate model and a controller is designed for an anti-lock braking system. Then the results show that though the system of a vehicle can be approximated roughly with the proposed model, good results can be obtained with the suggested controller.

The analysis testifies that RFPT can be applied in several areas successfully. The investigations also prove that the contributions suggested by the author improve the performance of the RFPT-based controllers and avoid several of their disadvantages. First it is shown in Chapter 5 that RFPT can be successfully applied in the field of chaos synchronization. Secondly the new structure proposed in Chapter 6 reduces the tracking errors achieved by the original versions of RFPT. Thirdly the two innovative methods introduced in Chapters 7 and 8 make the RFPT-based controllers stable. Fourthly in Chapters 9 and 10 two types of soft-computing-based controllers are combined with RFPT. Finally, in Chapters 11 and 12 a real and a possible aspect of real application of RFPT are shown: first, a hydrodynamic model of freeway traffic is analyzed in the viewpoint of stability and controlled with RFPT-based controller, then a simple model for an anti-lock braking system is designed and controlled without RFPT, but with the possibility of the extension. The constructed models make sure that there is a huge difference between the systems and their approximate representatives and that controlling them with RFPT-based controllers can bring results that reflect to reality.

The thesis is organized as follows. First, in Chapter 2 some classical controllers are reviewed that use Lyapunov function for control or parameter tuning. In Chapter 3 some nonlinear systems are introduced that are applied to help the analysis of the worked out methods by simulations. In Chapter 4 the basics of Robust Fixed Point Transformations are shown. In Chapter 5 the effectiveness of RFPT in chaos synchronization is investigated. Chapters 6-8 contain the improving extensions for RFPT: the new structure, the fuzzy-like parameter tuning and the VS-type stabilization method, respectively. Chapters 9-10 present two prevalent soft computing-based families of controllers that are extended with RFPT. In Chapters 11 and 12 two realistic phenomenons are modeled, controlled, and parsed: in the prior with-, in the latter without RFPT (but with the possibility of the extension). The last chapter deals with the final conclusions.

2

State of the art

In Chapter 1 the development of nonlinear control theory and some open questions of the field are summarized. In this chapter, the emphasis is put on those methods that form the basis for the new ideas of this work. First, Lyapunov Stability Theory is briefly explained, then some traditional controllers are introduced and finally, two soft computing techniques are detailed that can advantageously be used for control purposes.

2.1 Lyapunov Stability Theory

In the first part of the 19^{th} century, stability of nonlinear systems was a problematic subject for the scientists. Only a few results were at hand to answer the question weather a system is stable or not. The first major aid came from Aleksandr Lyapunov in 1892 [26, 27]. In his dissertation he introduced his stability theory and an approach called Lyapunov's second or "direct" method in which he showed a way how to determine a nonlinear system's stability without solving its equations of motion. Since most of the problems appearing in real life do not have analytical solutions in closed form and the numerical solutions are valid only with time limitation, Lyapunov's method brought a breakthrough for the field of control. His development proved to be so significant that the stability of most of the controllers is ensured by his "direct" method even in our days.

Assume a dynamic system described by a set of ordinary differential equations expressing by arrays as

$$\dot{x} = f\left(x, t\right) \tag{2.1}$$

where $x \in \mathbb{R}^n$, $t \in [t_0, \infty)$, and $x\left(t_0\right) = x_0$. Let x denote some tracking error (in this case the main goal is to keep x as close to 0 as it is possible). If it is known that the system has one unique solution, according to Lyapunov Stability Theorem the followings can be stated:

- Point $x^* \in \mathbb{R}$ is an equilibrum point of the system if $\forall t \in [t_0, \infty)$ $f(x^*, t) = 0$.
- Equilibrum x^* is locally stable if every solution that starts close to x^*, remain close to x^* and asymptotically stable if in addition the solutions tend towards x^*.
- Equilibrum x^* is stable in $t = t_0$ if $\forall \epsilon > 0$ $\exists \delta(\epsilon, t_0) > 0$ such that $\|x(t_0) - x^*\| < \delta \Longrightarrow \|x(t) - x^*\| < \epsilon$, $\forall t > t_0$.
- Uniformly stable equilibrums can be defined if in the above definition δ depends only on ϵ.
- Equilibrum x^* is asimptotically stable at $t = t_0$ if it is stable and $\exists \delta(t_0) > 0$ such that $\|x(t_0) - x^*\| < \delta \Longrightarrow \|x(t) - x^*\| \to 0$ for $t \to \infty$.
- Equilibrum x^* is globally stable if it is stable for every initial condition $x_0 \in \mathbb{R}^n$.

To be able to determine whether system (2.1) is stable or not, there are two choices. The first option is integrating and solving (2.1) explicitly. This solution is applicable only in some special cases. Thanks to Lyapunov's direct method there is an other option. Instead of solving the equation, a uniformly continuous and positive definite function V can be constructed with a non-positive time-derivative on the domain $t \in [0, \infty)$ which can be used to prove the stability of the controller. This function can be calculated based on the tracking errors and the modeling errors of the system's parameters. According to the Barbalat lemma (stating that if a function $\frac{dV}{dt}$ is uniformly continuous and its integral $V(t)$ is bounded then the function itself converges to zero as $t \to \infty$) [28] the derivative of V converges to zero. As a result, the tracking errors and the modeling errors have to remain bounded or in a special case they have to converge to zero.

To show that function V is bounded, a function class $\mathcal{K}$ is introduced that can be used as upper and lower bounds of function V. Function $\kappa : [0, k) \to [0, \infty)$ where $k < \infty$ is a member of class $\mathcal{K}$ if $\kappa(0) = 0$ and $\kappa(t)$ is strictly increasing. Let us assume that $\alpha(\|x\|)$, $\beta(\|x\|)$, and $\gamma(\|x\|)$ belong to function class $\mathcal{K}$. In this case according to Lyapunov's second method it can be stated that

- If $V(0, t) = 0$ and $\forall x \in B_\epsilon(0)$ (where $B_\epsilon(0)$ denoted the ϵ vicinity of 0) and $\forall t \geq 0$: $V(x, t) \geq \alpha(\|x\|) > 0$ and $\dot{V}(x, t) \leq 0$ holds locally in x and for all t, then the equilibrum point $x = 0$ is locally stable.
- If $V(0, t) = 0$ and $V(x, t) \geq \alpha(\|x\|) > 0$ and $\dot{V}(x, t) \leq 0$ then the equilibrum point $x = 0$ is stable.
- If $V(0, t) = 0$ and $V(x, t) \geq \alpha(\|x\|) > 0$ and $\dot{V}(x, t) \leq 0$ and $V(x, t) \leq \beta(\|x\|) > 0$ then the equilibrum point $x = 0$ is uniformly stable.
- If $V(0, t) = 0$ and $V(x, t) \geq \alpha(\|x\|) > 0$ and $\dot{V}(x, t) \leq 0$ and $V(x, t) \leq \beta(\|x\|) > 0$, and $\dot{V}(x, t) \leq -\gamma(\|x\|)$ then the equilibrum point $x = 0$ is uniformly asymptotically stable.

2.2 Classical controllers

In this section, four classical controllers are discussed that are strongly related to the focus of this work. First the PID controller is shown in details which has been developed parallel with Lyapunov's method. Then the Computed Torque Control (CTC) and its special case, the Adaptive Inverse Dynamics are summarized together with a simple example. The example includes the proof of the controller's (Lyapunov) stability. Finally, the Model Reference Adaptive Controller, an illustrating example, and the Lyapunov stability proof are shown.

2.2.1 Proportional-Integral-Derivative Controller

The Proportional-Integral-Derivative (PID) controller was introduced in 1911 [29]. It was first used for automatic ship steering. It has become the most common feedback controller in the industry. In the industry, most of the machines are supervised by PID

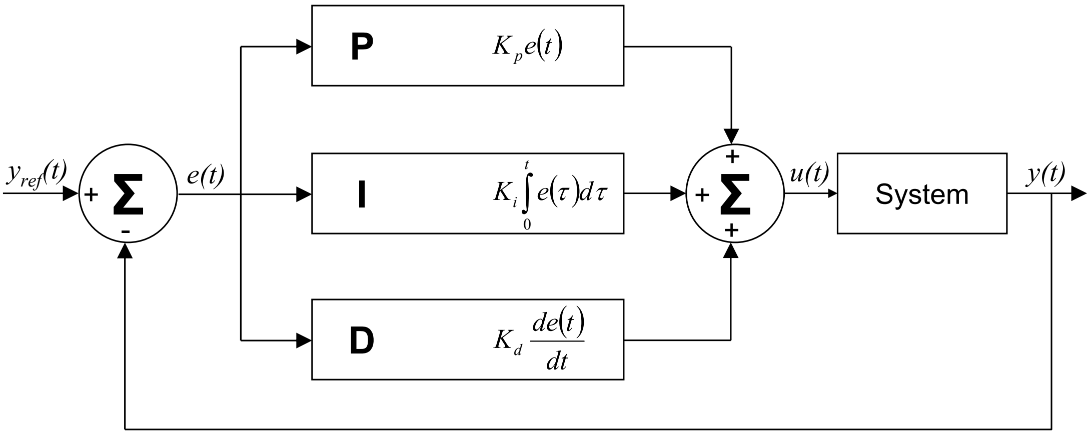


Figure 2.1: The block scheme of the traditional PID Controller.

controllers. This control strategy is popular because of its simplicity and easy handling. It can be described by

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \tag{2.2}$$

where u denotes the control signal, $e = y_{ref} - y$ marks the tracking error, y stands for the output and y_{ref} for the desired output of the controlled system; t is the time. K, T_i, and T_d denote the free variables of the controller. For simplicity the notations $K_p := K$, $K_i := \frac{K}{T_i}$, and $K_d := KT_d$ are also commonly used. The block diagram of the traditional PID controller is shown in Fig. 2.1.

Special cases of PID controller are also widely used nowadays. If one or two of the three parameters in (2.2) are set to zero, similar controllers can be gained like PD, PI, I, etc. controllers.

There are many possibilities for tuning the parameters of the PID controller. One of the most popular tuning strategies, called frequency response method has been developed by Ziegler and Nichols, see [30]. The essence of the tuning is the following: set K_i and K_d zero and increase K_p until the controlled system starts to oscillate. Let S_u denote this (high) value of K_p and P_u denote the oscillation period of the system. In this case, the proposed values for the parameters are $K_p = 0.6S_u$, $K_i = 2\frac{S_u}{P_u}$, and $K_d = \frac{S_u P_u}{8}$.

There are other possibilities how to determine the parameters, though they are not used any more in the industry. Instead of them PID tuning and loop optimization softwares are applied to ensure stable and good results (see e.g. [31]).

In the following, a simple example is shown how to determine numerically the parameters of a PID controller which controls a damped string described by the following equation:

$$\ddot{x} = -kx - b\dot{x} + cu \tag{2.3}$$

where x stands for the system state variable, k, b, and c are free parameters, and u denotes the control force. If a certain behavior is prescribed for the system by a reference model

$$\ddot{x}^{Ref} = -kx^{Ref} - b\dot{x}^{Ref} + cu \tag{2.4}$$

where the meaning of the parameters are the same, then the proper control force for the system can be calculated as

$$u = \frac{1}{c}\left(\ddot{x}^{Ref} + kx^{Ref} + b\dot{x}^{Ref}\right) \tag{2.5}$$

The PID correction can be added to (2.5) as

$$\begin{gathered} u = \tfrac{1}{c}\left(\ddot{x}^{Ref} + kx^{Ref} + b\dot{x}^{Ref}\right) + \\ P\left(x^{Ref} - x\right) + D\left(\dot{x}^{Ref} - \dot{x}\right) + I\int\limits_0^t \left(x^{Ref} - x\right) d\tau \end{gathered} \tag{2.6}$$

If (2.6) is substituted to (2.3) then after restructure the following equation is gained:

$$\ddot{x}^{Ref} - \ddot{x} = -(k + cP)(x^{Ref} - x) - (b + cD)(\dot{x}^{Ref} - \dot{x}) - cI\int\limits_0^t (x^{Ref} - x)d\tau \tag{2.7}$$

If $h = x^{Ref} - x$ denotes the error, then after a derivation (2.7) takes the form of

$$\dddot{h} = -(k + cP)\dot{h} - (b + cD)\ddot{h} - cIh \tag{2.8}$$

Let $h = e^{\alpha t}$. In this case,

$$\alpha^3 = -(k + cP)\alpha - (b + cD)\alpha^2 - cI \tag{2.9}$$

from which

$$\alpha^3 + (k + cP)\alpha + (b + cD)\alpha^2 + cI = 0 \tag{2.10}$$

is gained. Let α_1, α_2, and α_3 denote the roots of (2.10). It can be stated that

$$\begin{aligned} k + cP &= \alpha_1\alpha_2 + \alpha_2\alpha_3 + \alpha_1\alpha_3 \\ cI &= -\alpha_1\alpha_2\alpha_3 \\ b + cD &= -\alpha_1 - \alpha_2 - \alpha_3 \end{aligned} \tag{2.11}$$

In the knowledge of α_1, α_2, and α_3, the parameters of the PID controller can be set so that (2.8) converges to 0.

Despite the popularity of the PID controllers, they can be applied only if the controlled system is transparent and the effects of the feedback of the PID controller can be followed qualitatively. If not, then the controller cannot achieve optional system behavior. For example, if the pendulum of a cart-pendulum system (see Chapter 3) passes through the horizontal line, the behavior of the system changes and the control law determined by the PID controller will not be valid for the system any more. As an example for proper systems, the damped strings could be mentioned, because they are qualitatively transparent and they can be used as approximate models for many control problems, e.g. for stabilization tasks around an operating point.

2.2.2 Computed Torque Control (CTC) and the adaptive inverse dynamics control

The Computed Torque Control [32] is a control strategy usually applied on robots. The most important property of relatively simple robots is that they can be described analytically, so a relationship can be established between the joint coordinate accelerations and the torques or forces acting on the system (the forces and torques are made partly by the robot's own drives and/or by its environment with which the system may be in dynamic coupling). The relationship is described by the so-called Euler-Lagrange equations of motion:

$$\mathbf{H}(\mathbf{q})\frac{d^2\mathbf{q}}{dt^2} + \mathbf{h}\left(\mathbf{q}, \frac{d\mathbf{q}}{dt}\right) = \mathbf{Q} \tag{2.12}$$

where $\mathbf{H}(\mathbf{q})$ denotes the inertia matrix of the system, a part of $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is quadratic in $\dot{\mathbf{q}}$ and describes e.g. the Coriolis terms, while its other part depending only on $\mathbf{q}$ is responsible for the gravitational effects. Due to physical reasons $\mathbf{H}$ is always symmetric and positive definite. The term $\mathbf{Q}$ stands for the generalized forces of the robot's own drives and the environment, e.g. forces for the prismatic generalized coordinates, and torques for the rotational axes. In the possession of this model (on the basis of purely kinematic considerations) some desired $\frac{d^2\mathbf{q}^{\mathbf{des}}}{dt^2}$ can be computed in each control cycle to exert the necessary $\mathbf{Q}^{\mathbf{des}}$. This part of the controller is often referred to as "feedforward" control. For more precise tracking the "feedforward part" generally has to be completed by PID-type feedback terms based on the tracking error.

An important practical problem of CTC is that in many cases it is very difficult (or even impossible) to identify the parameters of the analytical models of the systems (see e.g. the model for the six degree of freedom PUMA robot [33], where the model construction took five weeks for three persons). Another practical problem in the application of this method is that normally there are no sensors available that could exactly measure the external (e.g. environmental) parts that affect $\mathbf{Q}$. Their effects can be observed only subsequently and generally cannot efficiently be compensated by simply prescribing some feedback correction in $\frac{d^2\mathbf{q}^{\mathbf{des}}}{dt^2}$.

If the kinematic model of the system is precisely known, the Adaptive Inverse Dynamics Control can be a solution. Let $\mathbf{p}$ represent the dynamical parameters (unknown) and $\mathbf{Y}\left(\mathbf{q}, \frac{d\mathbf{q}}{dt}, \frac{d^2\mathbf{q}}{dt^2}\right)$ the array built up based on the kinematic functions (known). The dynamic model can be formulated as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}\left(\mathbf{q}, \dot{\mathbf{q}}\right) = \mathbf{Q} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{p} \tag{2.13}$$

It is supposed that some approximation for $\mathbf{H}(\mathbf{q})$, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{p}$ are available as $\hat{\mathbf{H}}(\mathbf{q})$, $\hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$, and $\hat{\mathbf{p}}$. The exerted forces may contain feedback-correction depending on the tracking error and its derivatives $\mathbf{e} = \mathbf{q}^{\mathbf{des}} - \mathbf{q}$, $\dot{\mathbf{e}} = \dot{\mathbf{q}}^{\mathbf{des}} - \dot{\mathbf{q}}$, and $\ddot{\mathbf{e}} = \ddot{\mathbf{q}}^{\mathbf{des}} - \ddot{\mathbf{q}}$, with some symmetric positive definite gain matrices $\mathbf{K_0}$ and $\mathbf{K_1}$. In this case

$$\hat{\mathbf{H}}(\mathbf{q})(\ddot{\mathbf{q}}^{\mathbf{des}} + \mathbf{K_0}\mathbf{e} + \mathbf{K_1}\dot{\mathbf{e}}) + \hat{\mathbf{h}}\left(\mathbf{q}, \dot{\mathbf{q}}\right) = \mathbf{Q} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}\left(\mathbf{q}, \dot{\mathbf{q}}\right) \tag{2.14}$$

It is assumed that $\mathbf{Q}$ originates from the drives and does not contain unknown external components, so by subtracting (2.14) from (2.13) we can obtain

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}\left(\mathbf{q}, \dot{\mathbf{q}}\right) - \hat{\mathbf{H}}(\mathbf{q})(\ddot{\mathbf{q}}^{\mathbf{des}} + \mathbf{K_0}\mathbf{e} + \mathbf{K_1}\dot{\mathbf{e}}) - \hat{\mathbf{h}}\left(\mathbf{q}, \dot{\mathbf{q}}\right) = 0 \tag{2.15}$$

and then by subtracting with $\hat{\mathbf{H}}\left(\mathbf{q}\right)\ddot{\mathbf{q}}$ and reordering

$$\begin{gathered}\hat{\mathbf{H}}(\mathbf{q})(\ddot{\mathbf{e}}+\mathbf{K_0}\mathbf{e}+\mathbf{K_1}\dot{\mathbf{e}}) = \\ \Big(\mathbf{H}(\mathbf{q})-\hat{\mathbf{H}}(\mathbf{q})\Big)\,\ddot{\mathbf{q}}+\Big(\mathbf{h}\left(\mathbf{q},\dot{\mathbf{q}}\right)-\hat{\mathbf{h}}\left(\mathbf{q},\dot{\mathbf{q}}\right)\Big) = \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\left(\mathbf{p}-\hat{\mathbf{p}}\right)\end{gathered} \tag{2.16}$$

where the left hand side contains the model data, while the other side contains the modeling errors: $\tilde{\mathbf{H}} := \mathbf{H}(\mathbf{q})-\hat{\mathbf{H}}(\mathbf{q})$, $\tilde{\mathbf{h}} := \mathbf{h}\left(\mathbf{q},\dot{\mathbf{q}}\right)-\hat{\mathbf{h}}\left(\mathbf{q},\dot{\mathbf{q}}\right)$, and $\tilde{\mathbf{p}} := \mathbf{p}-\hat{\mathbf{p}}$. Via multiplying both sides with the inverse of the known model, the following standard form is obtained:

$$\begin{bmatrix}\dot{\mathbf{e}}\\ \ddot{\mathbf{e}}\end{bmatrix}-\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{K_0} & -\mathbf{K_1}\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{\Phi}\tilde{\mathbf{p}}\end{bmatrix} \tag{2.17}$$

where $\mathbf{\Phi} = \hat{\mathbf{H}}^{\mathbf{-1}}\left(\mathbf{q}\right)\mathbf{Y}\left(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}\right)$.

Let us introduce the following notations: $\mathbf{x} := \begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix}$, $\dot{\mathbf{x}} := \begin{bmatrix}\dot{\mathbf{e}}\\ \ddot{\mathbf{e}}\end{bmatrix}$, $\mathbf{B} := \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}$, and $\mathbf{A} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{K_0} & -\mathbf{K_1}\end{bmatrix}$. By this, the system can be described in a more simple form:

$$\dot{\mathbf{x}}-\mathbf{A}\mathbf{x}=\mathbf{B}\mathbf{\Phi}\tilde{\mathbf{p}} \tag{2.18}$$

For the tracking error $\mathbf{e}$, the first derivative of it $\dot{\mathbf{e}}$, and the parameter estimation error $\tilde{\mathbf{p}}$ the following Lyapunov function V can be constructed:

$$V=\mathbf{x}^{\mathbf{T}}\mathbf{P}\mathbf{x}+\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{R}\tilde{\mathbf{p}} \tag{2.19}$$

where $\mathbf{P}$ and $\mathbf{R}$ are constant, symmetric positive definite matrices. In this case

$$\dot{V}=\dot{\mathbf{x}}^{\mathbf{T}}\mathbf{P}\mathbf{x}+\mathbf{x}^{\mathbf{T}}\mathbf{P}\dot{\mathbf{x}}+\dot{\tilde{\mathbf{p}}}^{\mathbf{T}}\mathbf{R}\tilde{\mathbf{p}}+\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{R}\dot{\tilde{\mathbf{p}}}<0 \tag{2.20}$$

From (2.18) it follows that

$$\dot{V}=\mathbf{x}^{\mathbf{T}}\left(\mathbf{A}^{\mathbf{T}}\mathbf{P}+\mathbf{P}\mathbf{A}\right)\mathbf{x}+\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{\Phi}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{P}\mathbf{x}+\mathbf{x}^{\mathbf{T}}\mathbf{P}\mathbf{B}\mathbf{\Phi}\tilde{\mathbf{p}}+\dot{\tilde{\mathbf{p}}}^{\mathbf{T}}\mathbf{R}\tilde{\mathbf{p}}+\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{R}\dot{\tilde{\mathbf{p}}}<0 \tag{2.21}$$

Due to the symmetry of matrices $\mathbf{P}$ and $\mathbf{R}$ (2.21) can be simplified as

$$\dot{V}=\mathbf{x}^{\mathbf{T}}\left(\mathbf{A}^{\mathbf{T}}\mathbf{P}+\mathbf{P}\mathbf{A}\right)\mathbf{x}+\mathbf{2}\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{\Phi}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{P}\mathbf{x}+\mathbf{2}\tilde{\mathbf{p}}^{\mathbf{T}}\mathbf{R}\dot{\tilde{\mathbf{p}}}<0 \tag{2.22}$$

To guarantee $dV/dt<0$ for any finite $\mathbf{x}$, the following restrictions can be prescribed: let $\mathbf{U}$ be a negative definite symmetric matrix, and let

$$\mathbf{A^T P} + \mathbf{PA} = \mathbf{U} \tag{2.23}$$

further

$$\tilde{\mathbf{p}}^{\mathbf{T}} \left(\mathbf{\Phi^T B^T P x} + \mathbf{R}\dot{\tilde{\mathbf{p}}} \right) = \mathbf{0} \Rightarrow \dot{\tilde{\mathbf{p}}} = -\mathbf{R^{-1} \Phi^T B^T P x} \tag{2.24}$$

where (2.23) is referred as Lyapunov equation. Normally an appropriate **U** is prescribed and the task is to find a proper **P** for this **U** by solving the Lyapunov Equation. The Lyapunov Equation sets linear functional connection between the elements of **P** and **U** that may or may not have solution. (For the existence of a solution the real part of each eigenvalue of **A** must be negative.) Since **A**=constant, the Lyapunov Equation has to be solved only once in order to find a proper **P** for the prescribed **U**. To satisfy the second constraint (2.24), its right hand side has to be expressed from its definition through **B** and **Φ**. It is obtained that

$$\dot{\tilde{\mathbf{p}}} = \dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}} = -\mathbf{R^{-1} Y \hat{H}^{-1}} \left[\mathbf{0}, \mathbf{I}\right] \mathbf{Px} \tag{2.25}$$

in which the computational burden mainly consists of the need for inverting the model inertia matrix that must have the exact, intricate form determined by the particular kinematic model of the given system.

If the adaptation rule is applied, then the following cases can be separated:

- If $\|\mathbf{x}\| \to 0$ and $\|\tilde{\mathbf{p}}\| > F > 0$ then exponential trajectory tracking is achieved without exactly learning the system model.

- If $\|\mathbf{x}\| \to 0$ and $\|\tilde{\mathbf{p}}\| \to 0$ then exponential trajectory tracking is achieved with exactly learnt system model.

$\|\mathbf{x}\| > E > 0$ for arbitrarily long time is not possible since an initially finite positive value $V(0)$ with at least constant speed of decrease has to achieve 0 during finite time.

2.2.3 Model Reference Adaptive Controller

The Model Reference Adaptive Controller (MRAC) belongs to the family of direct adaptive controllers [34]. It was proposed in 1958 to control an aircraft driven by a joystick [35]. It had stability problems during the first few years until Lyapunov functions have been started to be used for the design. The first successes were reached in 1966 [36, 37].

MRAC is based on the idea of constructing a reference model that determines the desired behavior of the system. Then the control signal is calculated by the difference of the model's and the system's output (tracking error). Its structure is very simple, having four main parts:

- The system: it has known structure but contains unknown parameters. For nonlinear systems it can be said that the structure of the nonlinear equations are known, but some of the parameters are not.
- The reference model: it specifies the desired output of the controlled system. It has to be designed parallel with the controller. Further, it has to reflect the performance specification (like rise and settling time, overshoot, etc.) and its output has to be achievable for the system (e.g. its order and relative degree have to match the system's assumed order and degree).
- The feedback controller (or control law): it is parameterized by adjustable parameters. If the system parameters are all known, it has to force the system to act exactly like the reference model (prefect tracking). If the system parameters are not known, it has to achieve perfect tracking asimptotically.
- An adaptation law: it is used to adjust the parameters of the controller. The goal is to set the parameters so that the system's output equals the model's output. The main issue of the MRAC design is to ensure that the controller remains stable and the tracking error converges to zero.

The block scheme of the traditional MRAC is shown in Fig. 2.2.

For designing a Model Reference Adaptive Controller, as an example, consider the system assumed in [5]. In this very simple system a mass m is settled on a frictionless surface and controlled by a motor: $m\ddot{x} = u$, where u denotes the force of the motor and x is the position of the mass. The positioning commands $r(t)$ come through a joystick handled by a human. In [5] the following reference model is suggested: $\ddot{x}^m + \lambda_1 \dot{x}^m + \lambda_2 x^m = \lambda_2 r(t)$, where x^m is the output of the reference model. Parameters λ_1 and λ_2 ($\lambda_1, \lambda_2 > 0$) are chosen to reflect the performance specifications of the systems.

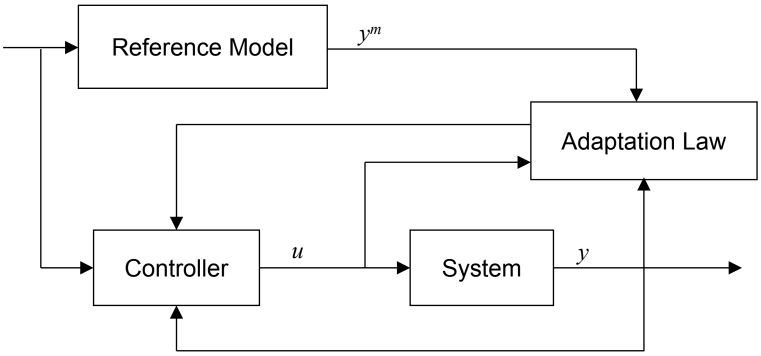


Figure 2.2: The block scheme of the traditional Model Reference Adaptive Controller taken from [38].

If the mass m is known, perfect tracking is achieved by the control law $u = m\left(\ddot{x}^m - \lambda\dot{e} - \lambda^2 e\right)$, where $e = x(t) - x^m(t)$ and λ is strictly positive. So the exponentially convergent tracking error dynamics are $\ddot{e} + 2\lambda\dot{e} + \lambda^2 e = 0$.

If the mass is not known exactly, the applied control law is $u = \hat{m}\left(\ddot{x}^m - \lambda\dot{e} - \lambda^2 e\right)$, where $\hat{m}$ is adjustable. Let be $s = \dot{e} + \lambda e$, $v = \ddot{x}^m - 2\lambda\dot{e} - \lambda^2 e$, and $e_m = \hat{m} - m$. In this case, the error dynamics are $m\dot{s} + \lambda m s = e_m v$. The parameter adjustment for the mass is $\dot{\hat{m}} = -\gamma v s$, where γ is called the adaptation gain, and it is a positive constant.

The stability analysis of the above explained controller can be shown by Lyapunov's theory. Consider the following function $V = \frac{1}{2}\left(ms^2 + \frac{1}{\gamma}e_m^2\right)$ as a Lyapunov function for the system, where $\dot{V} = -\lambda m s^2$. With the help of the Barbalat lemma it can be proven that s converges to zero which indicates the trajectory and velocity tracking.

2.3 Soft computing techniques

In this section, two important soft computing techniques are summarized that can advantageously be used in the control area: the fuzzy theory and the field of neural networks. Both are relevant and getting more popular as the control tasks include more uncertainties and lack of knowledge.

2.3.1 Fuzzy Theory

Fuzzy control methodologies have emerged in recent years as promising tools to solve nonlinear control problems. The fuzzy approach was first proposed by Lotfi A. Zadeh, in 1965 when he presented his seminal paper on fuzzy sets [39]. Zadeh showed that fuzzy logic unlike classical logic can handle and interpret values between false (0) and true (1). One of the most successful application areas of Fuzzy Logic proved to be Fuzzy Logic Control (FLC), because FLC systems can replace humans for performing certain tasks, for example control of a power plant [1], or aeroelastic wing section [11], etc. [40, 41, 42].

An other significant reason for applying fuzzy techniques in control is their simple approach which provides to use heuristic knowledge for nonlinear control problem. In very complicated situations, where the plant parameters are subject to perturbations or when the dynamics of the systems are too complex to be described by exact mathematical models, adaptive schemes have to be used to gather data and adjust the control parameters automatically. Based on the universal approximation theorem [43] and by incorporating fuzzy logic systems into adaptive control schemes, a stable fuzzy adaptive controller is suggested in [44] which was the first controller being able to control unknown nonlinear systems. Afterwards, a wide variety of adaptive fuzzy control approaches have been developed for nonlinear systems, like [45, 46, 47]. In the following the basics of Fuzzy theory are summarized.

2.3.1.1 Definitions

Definition 1 (Universe). *A fuzzy universe is the domain of the observations. Values, objects that need to be classified.*

Definition 2 (Linguistic value). *Linguistic values are words, symbols (sets) defined by the rate of belonging of the elements of the universe.*

Definition 3 (Linguistic variable). *A linguistic variable is an overall notion with the help of which the linguistic values in a specific topic can be referred.*

Definition 4 (Membership function). *Membership function is a mapping expressing the rate of belonging of a universe element to a linguistic value.*

Definition 5 (Fuzzy set). *Fuzzy set is a set to the elements of which a number between 0 and 1 can be assigned. The assignment is the membership function. If A is a fuzzy set over universe X, then $\mu_A(x) : X \to [0,1]$ is the membership function of set A. In case of discrete sets $A = \sum\limits_{i=1}^{n} \mu_A(x_i)/(x_i)$ denotes the fuzzy set A, where x_is are elements of X, with $\mu_A(x_i)$ membership value (in set A). In continuous case the notation is $A = \int_X \mu_A(x)/x$, where $x \in X$ and $\mu_A(x)$ is its membership value in set A.*

Definition 6 (Height of a fuzzy set). *The height of fuzzy set A on universe X is $hgt(A) = \sup_{x \in X} \mu_A(x)$. The fuzzy sets with height=1 are called normalized fuzzy sets. The fuzzy sets with height<1 are called subnormal fuzzy sets.*

Definition 7 (Core). *The core of fuzzy set A on universe X is crisp subset of A: $core(A) = \{x \in X | \mu_A(x) = 1\}$.*

Definition 8 (Support). *The support of fuzzy set A on universe X is crisp subset of A: $supp(A) = \{x \in X | \mu_A(x) > 0\}$.*

Definition 9 (α-cut). *The α-cut of fuzzy set A on universe X is crisp subset of A: $\alpha - cut(A) = \{x \in X | \mu_A(x) \geq \alpha\}$. The core of A can also be defined as $core(A) = 1 - cut(A)$.*

Definition 10 (Strong α-cut). *The strong α-cut of fuzzy set A on universe X is crisp subset of A: $\overline{\alpha} - cut(A) = \{x \in X | \mu_A(x) > \alpha\}$. The support of A can also be defined as $supp(A) = \overline{0} - cut(A)$.*

Definition 11 (Convex fuzzy set). *The fuzzy set A on universe X is convex if $\forall x_1, x_2, x_3 \in X$, $x_1 \leq x_2 \leq x_3 \to \mu_A(x_2) \geq \min(\mu_A(x_1), \mu_A(x_3))$.*

Definition 12 (The normalization of a fuzzy set). *The normalization of fuzzy set A on universe X results in an other (normalized) fuzzy set A' for which $\mu_{A'}(x) = \frac{\mu_A(x)}{hgt(A)})$, $x \in X$.*

Definition 13 (Fuzzy subset). *The fuzzy set B is subset of fuzzy set A on universe X if $\forall x \in X \ \mu_A(x) \leq \mu_B(x)$*

Definition 14 (Fuzzy partition). *Fuzzy partition means the partitioning of the universe by linguistic variables. Let $A_1, A_2, ..., A_N$ denote fuzzy subsets of universe X so that $\forall x \in X \sum_{i=1}^{N_A} \mu_{A_i}(x) = 1$, where $A_i \neq \emptyset$, and $A_i \neq X$. In this case the set consist of fuzzy sets A_i is a fuzzy partition.*

Definition 15 (Fuzzy number). *The fuzzy set A on universe X (in most of the time $X = \mathbb{R}$) is a fuzzy number, if A is convex and normalized, $\mu_a(x)$ is semi-continuous and the core of A contains only one element.*

Definition 16 (Fuzzy interval). *The fuzzy set A on universe X is a fuzzy interval, if A is convex and normalized, and $\mu_a(x)$ is semi-continuous.*

2.3.1.2 Operations on fuzzy sets

The intersection and union operations defined by Zadeh in 1965 are the following. For the intersection:

$$\mu_{A\cap B} = \min\left(\mu_A(x), \mu_B(x)\right)$$

For the union:

$$\mu_{A\cup B} = \max\left(\mu_A(x), \mu_B(x)\right)$$

Since then, various definitions have been developed, like the T-norms, T-conorms, and the S-norms that all fulfill some given axioms.

Definition 17 (T-norm)**.** *T-norm T is a mapping $T : [0,1] \times [0,1] \to [0,1]$, with the following constraints:*

T-1 $T(a,1) = a$

T-2 $b \leq c \Rightarrow T(a,b) \leq T(a,c)$

T-3 $T(a,b) = T(b,a)$

T-4 $T\left(T\left(a,b\right),c\right) = T\left(a,T\left(b,c\right)\right)$

The T-norms also satisfy the following condition: $T_W(a,b) \leq T(a,b) \leq \min(a,b)$, where $T_W(a,b) = \begin{cases} a & b=1 \\ b & a=1 \\ 0 & \textit{otherwise} \end{cases}$ *is called Weber T-norm [48].*

Definition 18 (T-conorm)**.** *T-conorm S is a mapping $S : [0,1] \times [0,1] \to [0,1]$, with the following constraints:*

S-1 $S(a,0) = a$

S-2 $b \leq c \Rightarrow S(a,b) \leq S(a,c)$

S-3 $S(a,b) = S(b,a)$

S-4 $S\left(S\left(a,b\right),c\right) = S\left(a,S\left(b,c\right)\right)$

The T-conorms also satisfy the following condition: $\max(a,b) \leq S(a,b) \leq S_W$*, where*

$$S_W(a,b) = \begin{cases} a & b = 0 \\ b & a = 0 \\ 1 & otherwise \end{cases}$$ *is called Weber S-norm (T-conorm) [48].*

Definition 19 (Fuzzy complement)**.** *The complement defined by Zadeh is* $c(a) = 1 - a$*. The complement* $\overline{A}$ *of fuzzy set* A *can be defined by as follows*

c-1 $c(0) = 1$

c-2 $a > b \Rightarrow c(a) < c(b)$

c-3 $c\left(c\left(a\right)\right) = a$

Definition 20 (Fuzzy reasoning)**.** *Fuzzy logic can be deduced from fuzzy set theory just as classical logic from classical set theory. The operations "and", "or", and "not" correspond to "intersection", "union", and "complement", respectively. Fuzzy logic is built on sets enabling the predicates to be linguistic variables.*

Statement - *Fuzzy statements are simple statements with linguistic labels of fuzzy sets combined by "and", "or", and "not".*

Implication - *Fuzzy implications can be defined in many ways (just like in the classical case), but result in different outputs depending on the chosen T- and S-norms.*

2.3.1.3 Rule-based fuzzy reasoning

The block scheme of the rule-based fuzzy reasoning system can be seen in Fig. 2.3. The input (observation), and the output (conclusion) are usually not fuzzy-type quantity. The transformation between the crisp values and the fuzzy sets is made by the fuzzification, and defuzzification blocks. The deduction is made by the reasoning block using the a priori knowledge of the given rule base. The reasoning block determines how much each rule is valid for the concrete input. In case of multiply inputs the validity is determined by the "weakest" input. Then the conclusion is calculated.

Fuzzification - The fuzzifying block transforms a crisp input to a fuzzy set. The most often used technique is the singleton fuzzification (the membership function is 1 at the input, and 0 otherwise). Less simple, but closer to reality is if the uncertainty and accuracy of the input is illustrated and the input is transformed to e.g. a fuzzy number. The uncertainty can principally be represented by an α-cut.

Rule base - The rules give the basics of the rule-based fuzzy systems. The rule base describes the a priori knowledge on the system. The rules are usually "IF... THEN..." type rules. The i^{th} rule can be expressed as

$$R_i : \textbf{IF } x_1 \text{ is } X_{i,1} \text{ and } x_2 \text{ is } X_{i,2} \text{ and } \ldots \text{ and } x_n \text{ is } X_{i,n} \textbf{ THEN}$$
$$y_1 \text{ is } Y_{i,1} \text{ and } \ldots \text{ and } y_m \text{ is } Y_{i,m}.$$

where $x_1, ..., x_n$ are inputs with $X_{i,1}, ..., X_{i,n}$ linguistic values, $y_1, ..., y_m$ are output variables with $Y_{i,1}, ..., Y_{i,1m}$ linguistic values.

Reasoning - According to the reasoning strategy two main rule-based systems can be determined: the composition-based reasoning, which determines its output as the composition $X \circ R$; and the individual rule-based reasoning, which determines the output (Y') as the union of the composition of the inputs and the individual rules. The prior is the one based by fuzzy theory, but the latter provides less computational time this is why the individual rule-based reasoning is more prevalent.

Defuzzification - Defuzzification is responsible for the transformation of the fuzzy outputs to crisp values. Various methods are known depending on the output fuzzy set, but the most prevalent approaches are the center of area (CoA), the center of gravity (CoG), the center of maxima (CoM), and the mean of maxima

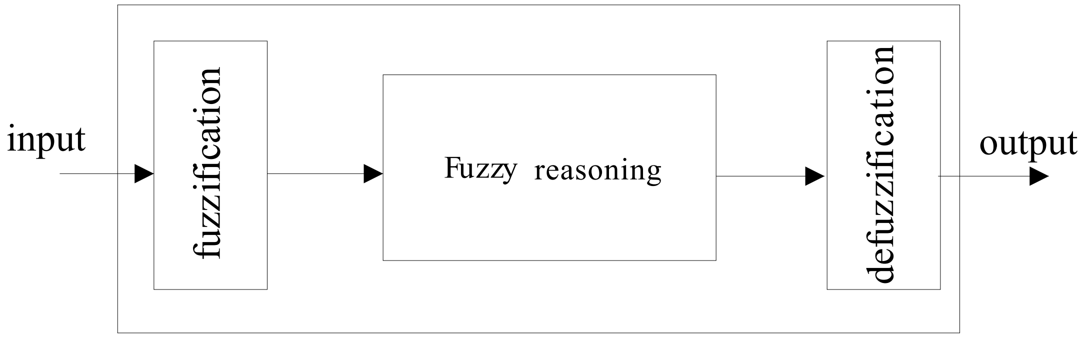


Figure 2.3: Fuzzy reasoning, taken from [49].

(MoM) defuzzification methods. As an example, the CoG defuzzification can be calculated as

$$y' = \frac{\int\limits_{Y'} \mu_{Y'}(y) y dy}{\int\limits_{Y'} \mu_{Y'}(y) dy} \tag{2.26}$$

for continuous values, and

$$y' = \frac{\sum\limits_{i=1}^{N_i} \mu_{Y'}(y_i) y_i}{\sum\limits_{i=1}^{N_i} \mu_{Y'}(y_i)} \tag{2.27}$$

for discrete values, where N_i denotes the number of the discrete values with the help of which $\mu_{Y'}(y)$ membership function can be discretized.

2.3.2 Artificial neural networks

Human recognition and control abilities far exceed those of complex intelligent control systems (e.g. robots). This has motivated scientist to analyze the human thinking to model neurons and nervous systems and use artificial neural networks in many areas (e.g. image precessing, signal processing, and control) [50]. The basic idea is according to natural neural networks to construct artificial systems (nets) consist of similar interconnected units (neurons). Though, the artificial neurons and neural networks are sketches compared to the natural ones, they have some important similar abilities, e.g. parallel processing, modularity, fault tolerance, and the ability to learn. The parallel synthesis shortens the computational time and makes sure that several disabled units do not influence the performance of the net considerably.

Neural networks are very helpful in classification, recognition problems, and optimization problems. In the thesis only feedback neural networks are used with backpropagation. The followings are valid mainly for this type of neural networks.

2.3.2.1 The structure of the neural networks

Neural networks are information processing tools characterized by parallel processing and a learning algorithm. Their unit is an artificial neuron with multiple inputs, one processing function, one output, and local memory. The easiest and most common type of neuron is a perceptron which calculates its output by a nonlinear transform of the weighted sum of the inputs (see Fig. 2.4)

$$y = f\left(\sum_{i=0}^{N} w_i x_i\right) = f\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right) \tag{2.28}$$

where $\mathbf{x} = [x_0, x_1, ..., x_N]^T$, $\mathbf{w} = [w_0, w_1, ..., w_N]$, x_i are input scalars with w_i weights, and the weighted sum is s. The value of x_0, called bias, is usually a nonzero constant. The nonlinear map is denoted by f, while y marks the output of the neuron. For determining the nonlinear map, many strategies can be found in the literature, like the binary transfer function

$$y(s) = \begin{cases} +1 & s > 0 \\ -1 & s \leq 0 \end{cases} \tag{2.29}$$

the piecewise-linear transfer function

$$y(s) = \begin{cases} +1 & s > 1 \\ s & -1 \leq s \leq 1 \\ -1 & s < -1 \end{cases} \tag{2.30}$$

and the sigmoid transfer function

$$y(s) = \frac{1 - e^{-Ks}}{1 + e^{-Ks}}; K > 0 \tag{2.31}$$

The three example functions are shown in Fig. 2.5.

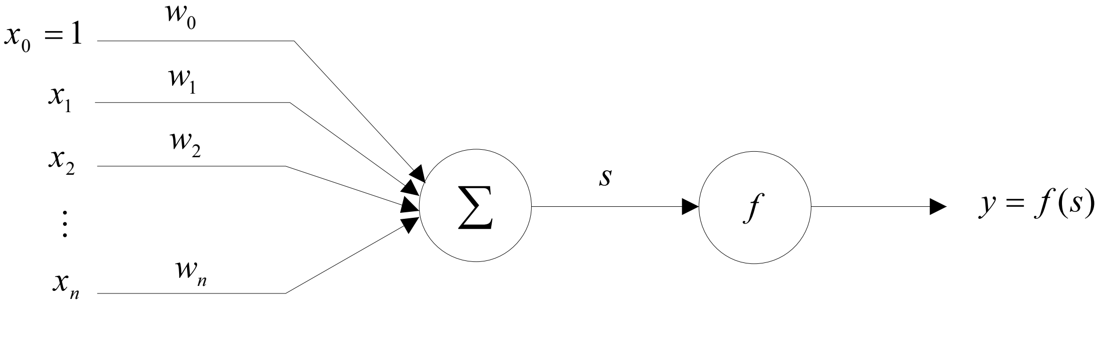


Figure 2.4: The scheme of a neuron without memory, with equal inputs, taken from [49].

2.3.2.2 Topology of the neural networks

The topology of a given neural network is how it is structured, e.g. where its in- and outputs are. The NNs are usually presented by directed graphs, where the nodes represent the neurons and the weighted edges denote the weighted connections. The neurons can be divided in three groups: input neurons (input of the network), output neurons (output of the network), and hidden neurons (inputs and outputs of other neurons in the network). They can be organized in layers, where each layer contains the same type of neurons. Thus, three different type of layers can be defined: input layer, output layer and hidden layer. The output of the input layer and the hidden layers are connected to other hidden layers or directly to the output layer. If the graph representation of the neural network contains a loop, it is called feedback neural network. Otherwise it is called feedforward neural network.

2.3.2.3 MultiLayer Perceptron

The most common multilayer feedforward neural network is the MultiLayer Perceptron (MLP) [51], where the connections are only between neighboring layers. The weights of the connections produce the free parameters of the NN. An example for an MLP is shown in Fig. 2.6. The example has $N+1$ inputs $(x_0^1, x_1^1, ..., x_n^1)$, two hidden layers with three and two neurons, and two outputs (y_1, y_2). The weight matrices are denoted by $\mathbf{W^{(1)}}$ and $\mathbf{W^{(2)}}$ while the biases for the layers are marked by x_0^1 and x_0^2. The applied transfer function is the sigmoid one.

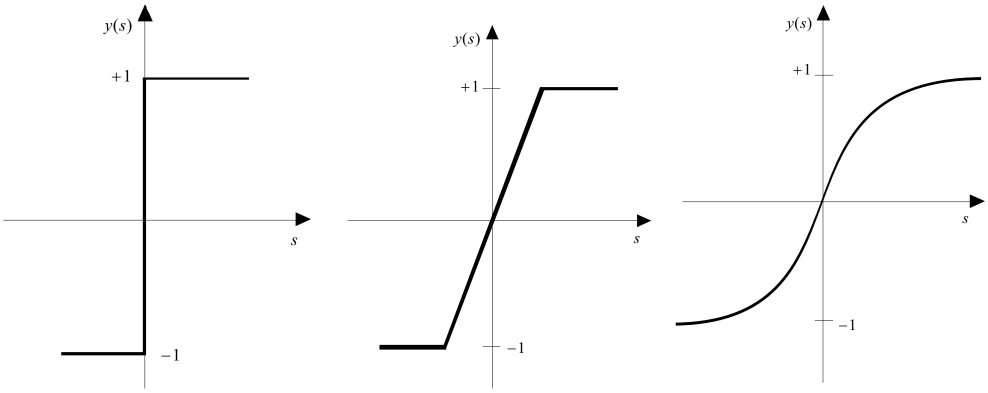


Figure 2.5: Typical nonlinearities in neurons: binary (left); piecewise-linear (middle); sigmoid (right).

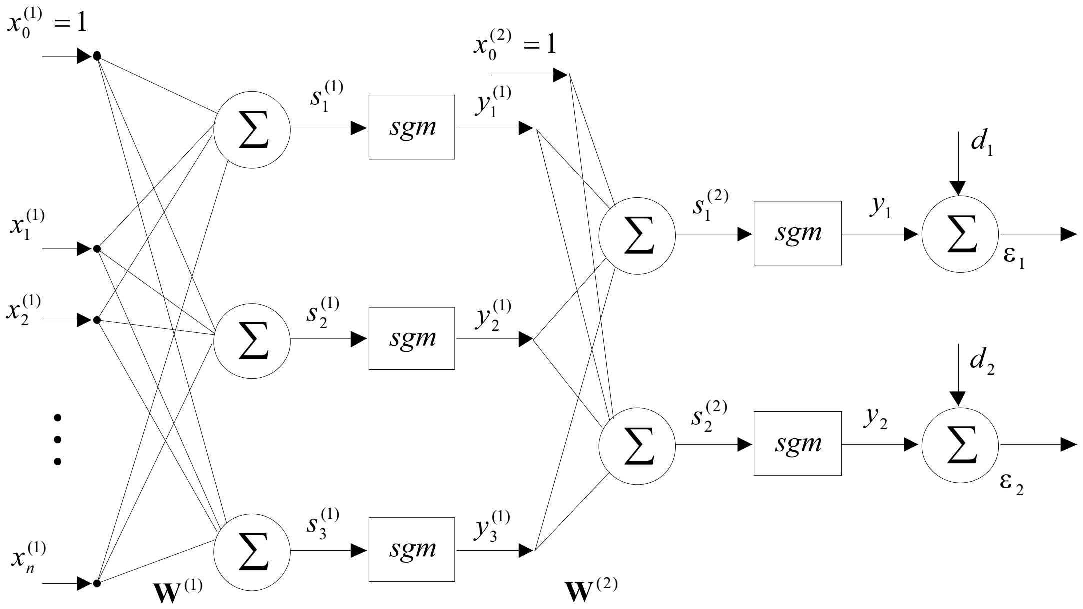


Figure 2.6: An example for a multilayer perceptron, taken from [49].

2.3.2.4 Supervised training

The desired behavior of the neural network is gained by the tuning of the weights which is called training. An appropriately complex neural network can be considered as an universal approximator, however achieving z optimal weight is an NP-complete problem to the training algorithms can give only near optimal results.

Figure 2.7 shows the general scheme of the training, where the expected coherent input-output pairs are given. In case of supervised training the output of the network can be compared to the desired output. From the comparison an error can be calculated which is used to modify the training in the proper way (through a criteria function or a parameter tuning algorithm).

2.3.2.5 Training one perceptron

The weight modification can be done by the least mean square (LMS) algorithm the criteria of which is the square of the actual error (see Fig. 2.8):

$$\begin{aligned} s &= \mathbf{w}^{\mathbf{T}}\mathbf{x} \\ y &= sgm(s) \\ \epsilon &= d - y \end{aligned} \tag{2.32}$$

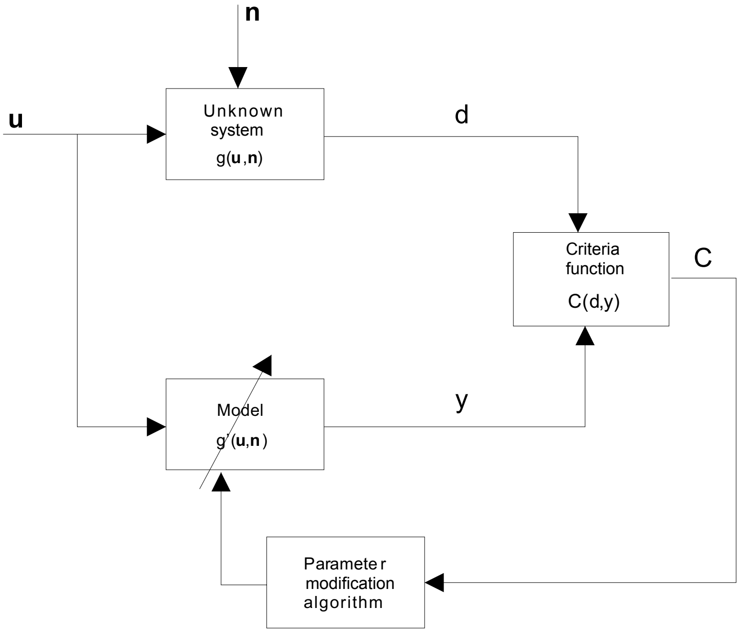


Figure 2.7: The block scheme of the training, where **u** the independent variables, **n** stands for the noise signals, and C marks the criteria function (usually a least mean square function), taken from [49].

where d is the output of the real system, y is the output of the network, and ϵ denotes the actual error. Function sgm stands for the nonlinear transfer function. The actual error can be expressed in more details:

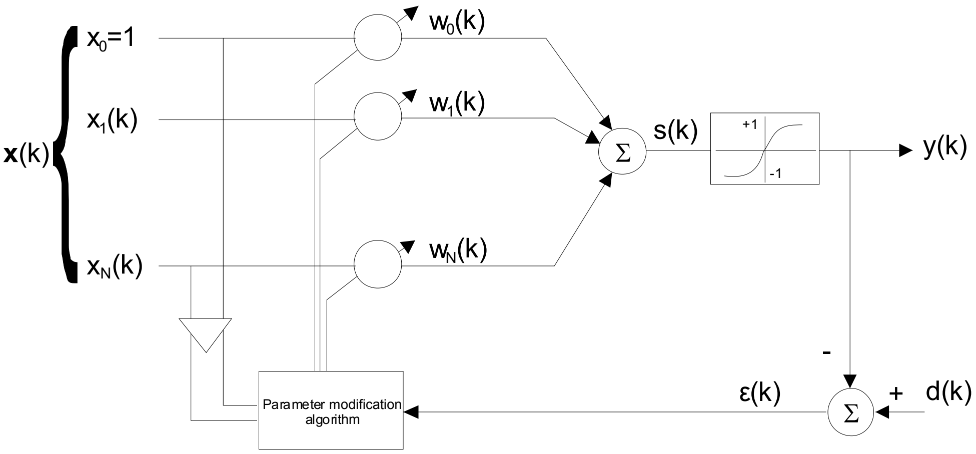


Figure 2.8: An illustrative example for modifying the weights of a neuron, taken from [49].

$$\epsilon(k) = d(k) - y(k) = d(k) - sgm\left(s\left(k\right)\right) = d(k) - sgm\left(\mathbf{w}^{\mathbf{T}}\left(k\right)\mathbf{x}\left(k\right)\right) \tag{2.33}$$

and the actual gradient can be determined as

$$\frac{\partial \epsilon^2}{\partial \mathbf{w}} = 2\epsilon\left(-sgm'\left(s\right)\right)\mathbf{x} \tag{2.34}$$

where **w** is the neuron's weight matrix. According to the gradient method the weight modification is the following:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu(k)\epsilon(k)sgm'\left(s\left(k\right)\right)\mathbf{x}(k) = \mathbf{w}(k) + 2\mu(k)\delta(k)\mathbf{x}(k) \tag{2.35}$$

where μ is the step size of the iteration.

2.3.2.6 The backpropagation training algorithm for the multilayer networks

The example for the backpropagation training algorithm is given for the network illustrated in Fig. 2.6. The network is a multilayer feedback NN which has two hidden layers with three and two neurons. The training algorithm is made with the coherent input-output $(\mathbf{x}, \mathbf{y})$ pairs and the gradient method. In this case the error can be determined as

$$\epsilon^2 = \epsilon_1^2 + \epsilon_2^2 = (y_1 - d_1)^2 + (y_2 - d_2)^2 \tag{2.36}$$

The actual gradient can be calculated just like in (2.34)

$$\frac{\partial \epsilon^2}{\partial w_{ij}^{(2)}} = -2\epsilon_1 sgm'\left(s_i^{(2)}\right)x_j^{(2)} = -2\delta_i^{(2)}x_j^{(2)} \tag{2.37}$$

$$\frac{\partial \epsilon^2}{\partial \mathbf{w}_i^{(2)}} = -2\epsilon_1 sgm'\left(s_i^{(2)}\right)\mathbf{x}^{(2)} = -2\delta_i^{(2)}\mathbf{x}^{(2)} \tag{2.38}$$

Thus, the weight modification is

$$\begin{gathered}\mathbf{w}_i^{(2)}(k+1) = \mathbf{w}_i^{(2)}(k) + 2\mu\epsilon_i(k)sgm'\left(s_i^{(2)}\right)\mathbf{x}^{(2)}(k) = \\ \mathbf{w}_i^{(2)}(k) + 2\mu\delta_i^{(2)}(k)\mathbf{x}^{(2)}(k)\end{gathered} \tag{2.39}$$

The error of the output of each neurons in the input layer are not known, but by applying the chain rule the derivatives can be determined (the weights of the input layer influence the neurons' linear and nonlinear outputs s and y, respectively, and through this the outputs of the other layers):

$$\begin{gathered}\frac{\partial\epsilon^2}{\partial w_{ij}^{(1)}}=\frac{\partial\epsilon^2}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}=\frac{\partial(\epsilon_1+\epsilon_2)^2}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}=\\=\frac{\partial\epsilon_1^2}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}+\frac{\partial\epsilon_2^2}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}=2\epsilon_1\frac{\partial\epsilon_1}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}+2\epsilon_2\frac{\partial\epsilon_2}{\partial s_i^{(1)}}\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}\end{gathered}\tag{2.40}$$

The error component is calculated as

$$\frac{\partial\epsilon_1}{\partial s_i^{(1)}}=\frac{\partial\epsilon_1}{\partial s_1^{(2)}}\frac{\partial s_1^{(2)}}{\partial y_i^{(1)}}\frac{\partial y_i^{(1)}}{\partial s_i^{(1)}}=-sgm'\left(s_1^{(2)}\right)w_{1i}^{(2)}sgm'\left(s_i^{(1)}\right)\tag{2.41}$$

$$\frac{\partial\epsilon_2}{\partial s_i^{(1)}}=\frac{\partial\epsilon_1}{\partial s_2^{(2)}}\frac{\partial s_2^{(2)}}{\partial y_i^{(1)}}\frac{\partial y_i^{(1)}}{\partial s_i^{(1)}}=-sgm'\left(s_2^{(2)}\right)w_{2i}^{(2)}sgm'\left(s_i^{(1)}\right)\tag{2.42}$$

By substituting (2.41) and (2.42) into (2.40), the following equation is gained:

$$\begin{gathered}\frac{\partial\epsilon^2}{\partial w_{ij}^{(1)}}=2\epsilon_1 sgm'\left(s_1^{(2)}\right)w_{1i}^{(2)}sgm'\left(s_i^{(1)}\right)\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}-\\-2\epsilon_2 sgm'\left(s_2^{(2)}\right)w_{2i}^{(2)}sgm'\left(s_i^{(1)}\right)\frac{\partial s_i^{(1)}}{\partial w_{ij}^{(1)}}=\\-2\left(\delta_1^{(2)}w_{1i}^{(2)}+\delta_2^{(2)}w_{2i}^{(2)}\right)sgm'\left(s_i^{(1)}\right)x_j^{(1)}=-2\delta_i^{(1)}x_j^{(1)}\end{gathered}\tag{2.43}$$

Thus, the weight modification is

$$\mathbf{w}_i^{(1)}(k+1)=\mathbf{w}_i^{(1)}(k)+2\mu\delta_i^{(1)}(k)\mathbf{x}^{(1)}(k)\tag{2.44}$$

2.4 Summary

In this chapter, those results are summarized that enable the arose of the direct antecedent of the thesis. First, the Lyapunov Stability Theory is explained then some traditional controllers are shown, like the PID controller, the Computed Torque Controller, the Adaptive Inverse Dynamics controller, and the Model Reference Adaptive Controller. Finally, the basics of two soft-computing techniques are presented: the fuzzy theory and the artificial neural networks.

3

Nonlinear systems

In the previous chapter, Lyapunov's stability theory, some classical controllers and some modern techniques are introduced that serve as basis for the research work presented in this dissertation. In this chapter, some of the well known nonlinear systems are shown. The new results introduced in the next chapters are analyzed with the help of these systems. First the FitzHugh-Nagumo neurons, the Duffing system, the Matsumoto-Chua circuit, and the Φ^6-type Van der Pol oscillator are shown, which are all chaotic systems. Then the cart-pendulum and the cart plus double pendulum systems are described. Finally, two realistic models are presented: the hydrodynamic model of freeway traffic and the Burckhardt tire model.

3.1 The FitzHugh-Nagumo neuron model

The FitzHugh-Nagumo (FHN) model comes from the four degree of freedom Hodgkin-Huxley model which was published in 1952 [52]. The FHN model, which was developed in 1961 [53], is a two degree of freedom reduction of the Hodgkin-Huxley model to be able to analyze it with phase plane techniques. It is used in Biology to model and control the electrical potential across cell membrane. The FHN model can be also used in chaos synchronization, where its dimensionless approximation is used for the investigations:

$$\begin{gathered}\dot{x}_1 = x_1(x_1 - 1)(1 - a_1 x_1) - x_2 - I + d_1 \\ \dot{x}_2 = b_1 x_1\end{gathered} \tag{3.1}$$

in which x_1 and x_2 are state variables, a_1 and b_1 are free parameters, I denotes the excitation of the system (responsible for the chaotic behavior) and d_1 interprets the disturbance force. If it is used for chaos synchronization, then a coupling element g_c (resistor) is included in the first equation and a second system is used:

$$\begin{gathered}
\dot{x}_1 = x_1(x_1-1)(1-a_1x_1) - x_2 - g_c(x_1-y_1) + I + d_1 \\
\dot{x}_2 = b_1x_1 \\
\dot{y}_1 = y_1(y_1-1)(1-a_2y_1) - y_2 - g_c(y_1-x_1) + I + u + d_2 \\
\dot{y}_2 = b_2y_1
\end{gathered} \tag{3.2}$$

where the second system gets an additional term u as control force. The variable $0 \leq g_c \leq 1$ determines the strength of the coupling between the two systems.

If the parameters are set to $a_1 = 10$, $a_2 = 5$, $b_1 = 1$, $b_2 = 0.5$, $I(t) = A_{exct}sin(2\pi f_{exct}t)$ where $A_{exct} = 0.5$, and $f_{exct} = 1.271\,Hz$, and the coupling element $g_c = 0.02$ is very low then Fig. 3.1 shows the phase space of the master systems and Fig. 3.2 illustrates the tracking error and system response tracking without control with $x_1 = 0.005$, $x_2 = 0.005$, $y_1 = -0.005$, and $y_2 = -0.005$ initial values.

In Chapter 5, based on [54] a RFPT-based PD controller is designed to synchronize the above systems with the given initial conditions. In [54] a fuzzy logic controller is used, but the synchronization is not achieved if the coupling element is small ($g_c < 0.5$).

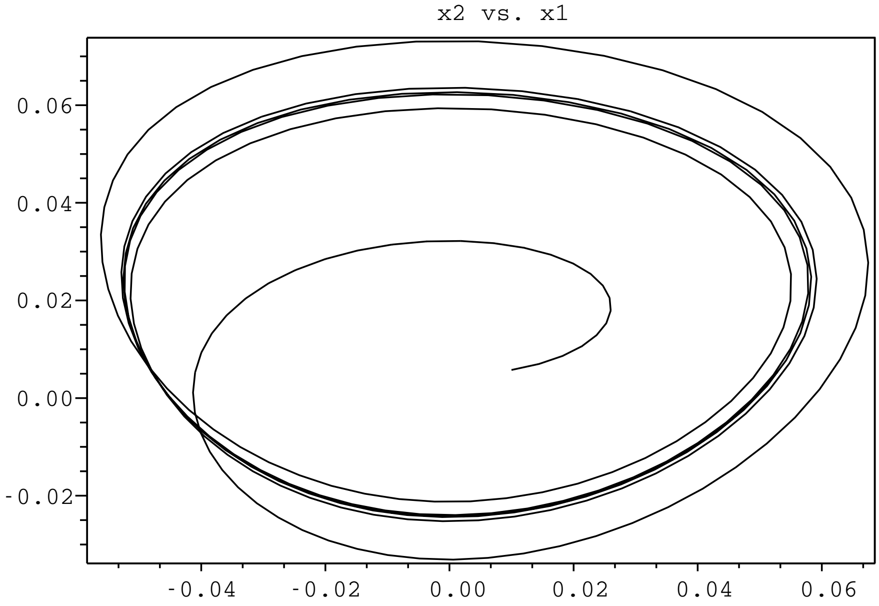


Figure 3.1: Phase space of the master FitzHugh-Nagumo system.

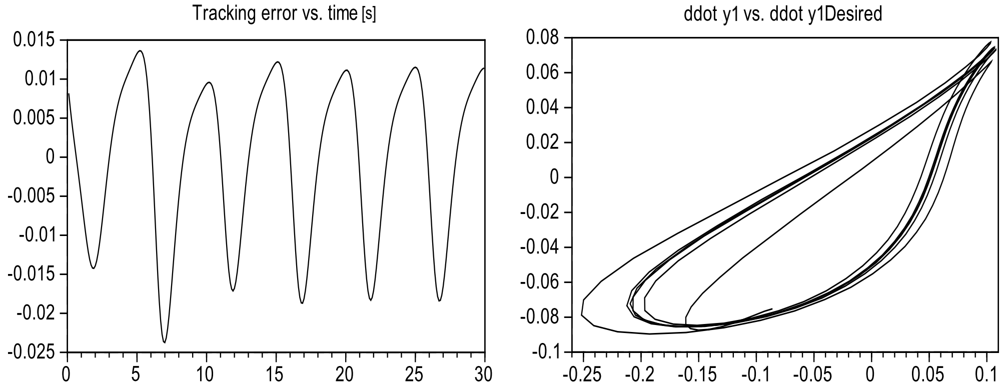


Figure 3.2: The tracking error (left) and system response tracking (right) of the FHN neurons without control, with $x_1 = 0.005$, $x_2 = 0.005$, $y_1 = -0.005$, and $y_2 = -0.005$ initial values. In case of perfect tracking, the right figure would contain one straight line.

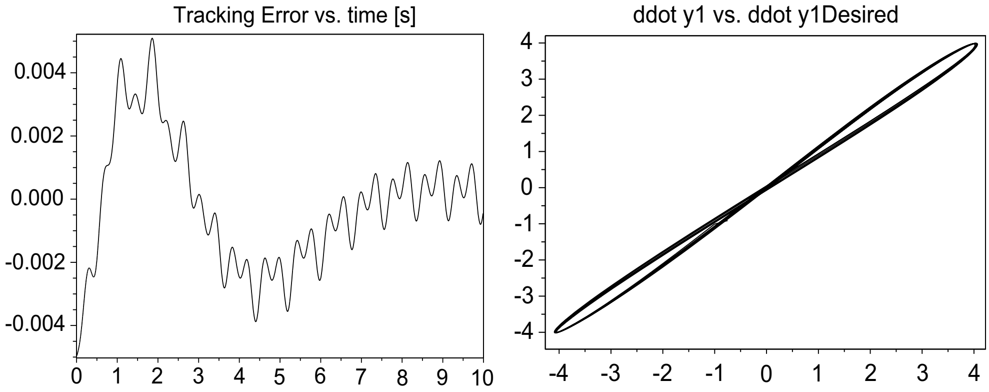


Figure 3.3: The tracking error (left) and system response tracking (right) of the FHN neurons without control, with $x_1 = 0$, $x_2 = 0$, $y_1 = 0.005$, and $Y_2 = 0$ initial values. In case of perfect tracking, the right figure would contain one straight line.

In Chapter 5, the simulations are successful even if the coupling element is very small ($g_c = 0.02$).

Figure 3.3 illustrates the tracking error of the first state variables and the response tracking with different initial values: $x_1 = 0$, $x_2 = 0$, $y_1 = 0.005$, and $y_2 = 0$. In Chapter 5, it is shown how the measurement noise and a simple filter can modify the RFPT's results during synchronizing two FHN systems with the above parameters and the new initial conditions.

3.2 The Matsumoto-Chua circuit

The Matsumoto-Chua circuit originates from the Chua circuit which was developed in 1983 [55] to demonstrate chaos as a robust physical phenomenon and to prove that the Lorenz attractor is chaotic in a rigorous mathematical sense. The Matsumoto-Chua circuit was developed based on the Chua circuit, one year later. It is a simple electronic circuit made of two capacitors, one linear resistor, one inductor, and only one nonlinear diode. The mathematical model used in this thesis is taken from [56]:

$$\begin{aligned} \dot{i}_L &= \tfrac{-1}{L_c} v_{C2} \\ \dot{v}_{C1} &= \tfrac{G}{C_1^{mc}} \left(v_{C2} - v_{C1}\right) - \tfrac{g_{mc}(v_{C1})}{C_1^{mc}} \\ \dot{v}_{C2} &= \tfrac{-G}{C_2^{mc}} \left(v_{C2} - v_{C1}\right) + \tfrac{i_L}{C_2^{mc}} \end{aligned} \tag{3.3}$$

in which C_1^{mc} and C_2^{mc} are two capacitors, L_c is the inductance of a coil, G represents the reciprocal value of a common resistor while $g_{mc}\left(v_{C1}\right)$ describes the nonlinear element with the characteristics as follows

$$\begin{gathered} \texttt{if } v_{C1} <= -1 \texttt{ then } g_{mc} = -S_{big} + S_{small}\left(v_{C1}+1\right) \\ \texttt{if } v_{C1} > -1 \texttt{ and } v_{C1} < 1 \texttt{ then } g_{mc} = S_{big} v_{C1} \\ \texttt{if } v_{C1} >= 1 \texttt{ then } g_{mc} = S_{big} + S_{small}\left(v_{C1}-1\right). \end{gathered} \tag{3.4}$$

This system has three degree of freedom (variables i_L: current of the inductance, v_{C1}: voltage of capacitor C_1^{mc}, and v_{C2}: voltage on capacitor C_2^{mc}).

For chaos synchronization a second system can be introduced which has the same structure as the master system (but it has different parameters), with a little modification (see Fig. 3.4): a control current i_u is introduced by a current generator in parallel position of capacitor C_2^{mc} resulting in the following equations of motion:

$$\begin{aligned} \dot{\tilde{i}}_L &= \tfrac{-1}{\tilde{L}_c} \tilde{v}_{C2} \\ \dot{\tilde{v}}_{C1} &= \tfrac{\tilde{G}}{\tilde{C}_1^{mc}} \left(\tilde{v}_{C2} - \tilde{v}_{C1}\right) - \tfrac{\tilde{g}_{mc}(\tilde{v}_{C1})}{\tilde{C}_1^{mc}} \\ \dot{\tilde{v}}_{C2} &= \tfrac{-\tilde{G}}{\tilde{C}_2^{mc}} \left(\tilde{v}_{C2} - \tilde{v}_{C1}\right) + \tfrac{\tilde{i}_L - i_u}{\tilde{C}_2^{mc}} \end{aligned} \tag{3.5}$$

where the symbol $\sim$ denotes the slave system variables and parameters with the same meaning, respectively.

If the parameters are taken from [56] as follows: $C_1^{mc} = 1/10\,F$, $C_2^{mc} = 2\,F$, $L_c = 1/7\,H$, $G = 0.7\,\frac{1}{\Omega}$, $S_{small} = -0.1\,\frac{1}{\Omega}$, $S_{big} = -4\,\frac{1}{\Omega}$, and the initial values are $v_{C1} = 1.45305\,V$, $v_{C2} = -4.36956\,V$, and $i_{L_c} = -0.15034\,A$, then the 3D view of the master

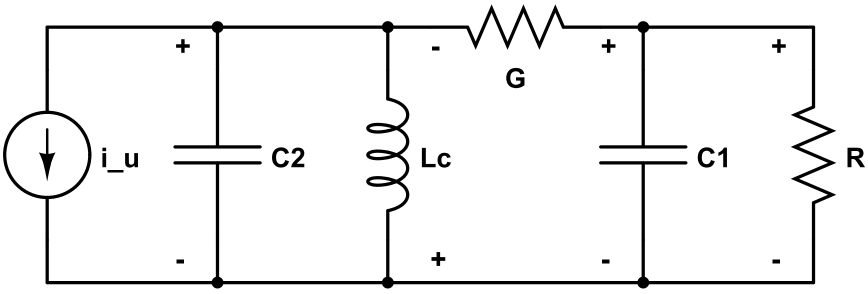


Figure 3.4: The Matsumoto-Chua circuit of [56] completed by a current generator of signal i_u for control purposes.

circuit's trajectory can be seen in Fig. 3.5, where the voltage of capacitor 2 v_{C2} provides the significant signal (the v_{C1} and i_L values are not directly controlled. They are regarded as characteristics of some coupled dynamical subsystems the states of which are out of any primary interest). Without control the realized system response ($\dot{\tilde{v}}_{C2}$) versus the desired response ($\dot{v}_{C2}$) can be compared in Fig. 3.6. Figure 3.7 shows the tracking error of the slave system.

In Chapter 5, successful chaos synchronization is made with two Matsumoto-Chua circuits based on an approximate model.

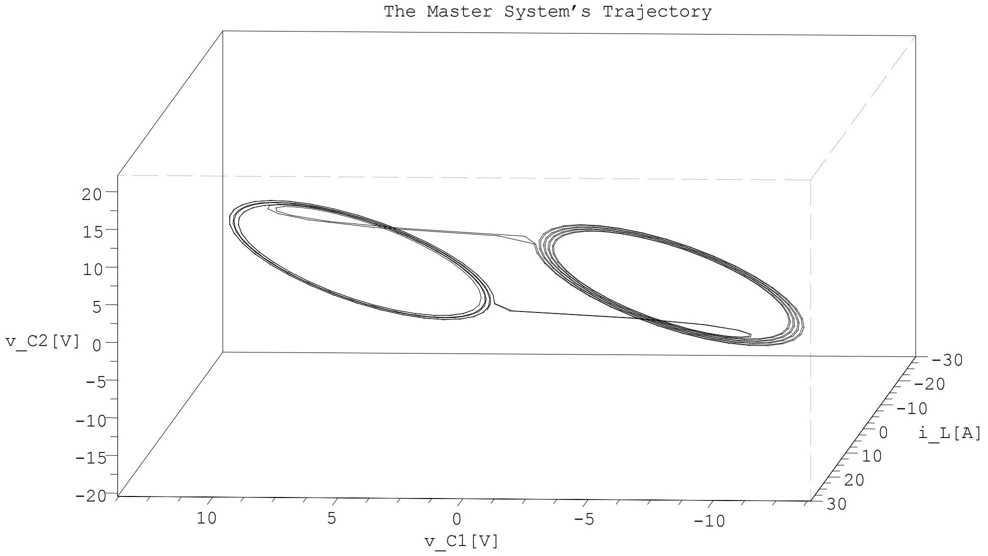


Figure 3.5: A 3D view of the chaotic trajectory produced by the master Matsumoto-Chua system.

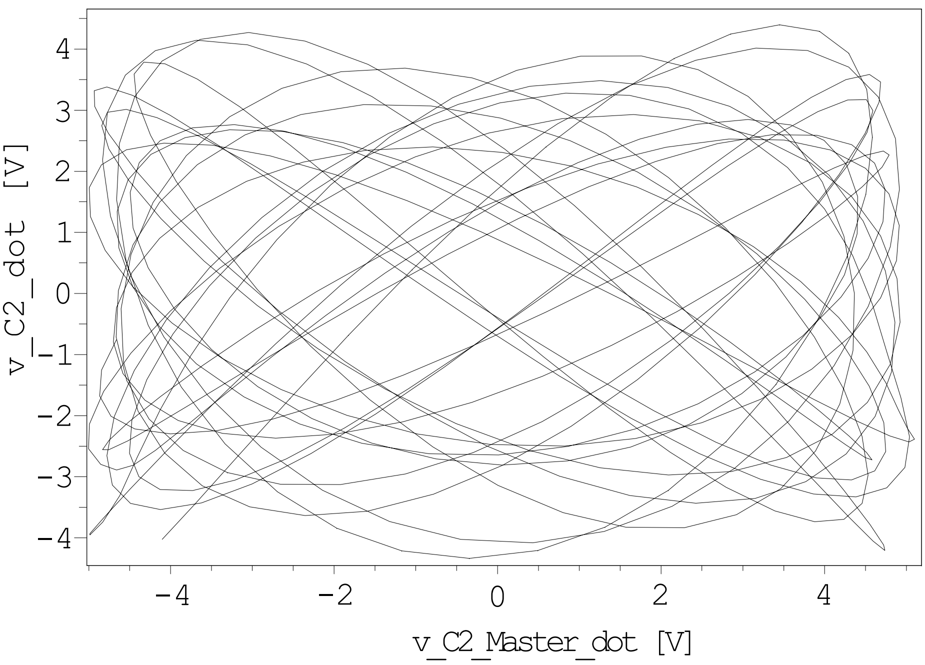


Figure 3.6: The realized system response ($\dot{\tilde{v}}_{C2}$ as v_C2_dot) versus the desired response ($\dot{v}_{C2}$ as v_C2_Master_dot) of the Matsumoto-Chua circuits without control. In ideal case one single straight line could be seen.

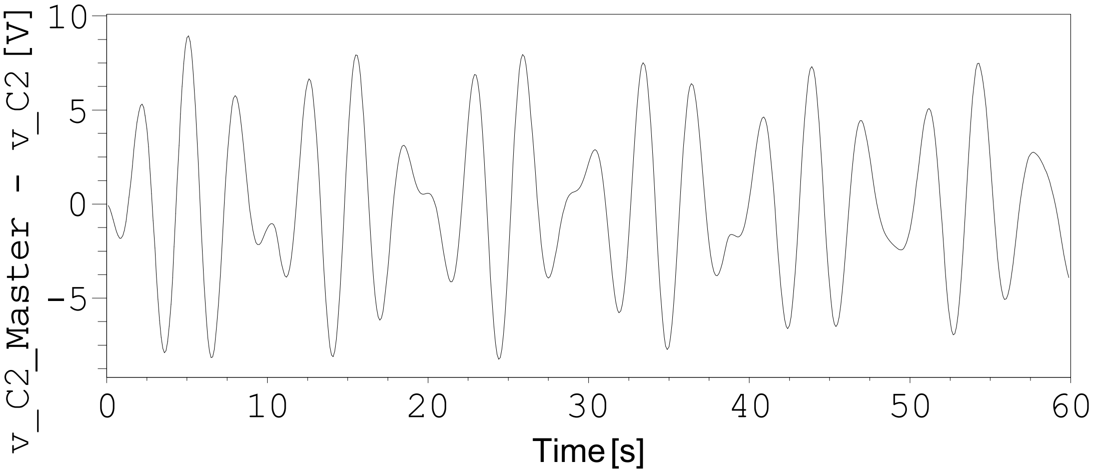


Figure 3.7: Tracking error $v_{C2} - \tilde{v}_{C2}$ of the Matsumoto-Chua circuits without control. The trajectories are strongly biased.

3.3 The Duffing System

The Duffing system originally comes from the nonlinear second order differential equation called Duffing equation [57] published by Georg Duffing in 1918 to model certain driven and damped oscillators (for example a spring pendulum). The system used in this dissertation follows his development:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\delta_1 x_2 + \alpha_1 x_1 - \beta_1 x_1^3 + a\cos\omega_d t + d_1 \end{aligned} \tag{3.6}$$

Here x_1 denotes the displacement, its first derivative (x_2) means the velocity, and the second derivative ($\dot{x_2}$) is responsible for the acceleration. The term $I = a\cos\omega_d t$ determines the external exciting force responsible for the chaotic motion of the system where a denotes the amplitude and ω_d is the frequency (they are free parameters). α (restoring force), β (non-linearity in the restoring force), and δ (damping) are also free parameters. The component d_1 can be interpreted as “disturbance force”.

The Duffing system is very well suitable for building chaos systems and for illustrating chaos synchronization problems. The main idea of the most approaches is to use the output of a master system to be followed by a slave system. The slave system has to be controlled so that its output tracks the master's one. In this case, the master system is described in (3.6), and the slave system is defined similarly:

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= -\delta_2 y_2 + \alpha_2 y_1 - \beta_2 y_1^3 + a\cos\omega_d t + u + d_2 \end{aligned} \tag{3.7}$$

where the meaning of the free parameters are the same, respectively, y_1 and y_2 are the state variables, and u denotes the control force.

The two systems have very similar structures, though depending on the parameters their behavior can be very different. If e.g. the following values are given to the parameters: $\omega_d = 2Hz$, $a = 1.2Nm$, $\alpha_1 = 1N$, $\alpha_2 = 0.8N$, $\beta_1 = 1\frac{N}{m^2}$, $\beta_2 = 1.5\frac{N}{m^2}$, $\delta_1 = 0.2N$, $\delta_2 = 0.3N$, and the initial values of the state variables are $y_1 = 3m$, $y_2 = 3\frac{m}{s}$, $x_1 = 2m$, and $x_2 = 2\frac{m}{s}$ then Fig. 3.8 shows the differing chaotic and non-synchronous motions of the two systems, while Fig. 3.9 describes the tracking error of the slave system. The figures illustrate how different two systems can behave having similar structures.

Further characteristic simulation results with these Duffing systems can be found in Chapter 6 where it is shown that if an extra controller is integrated into one of

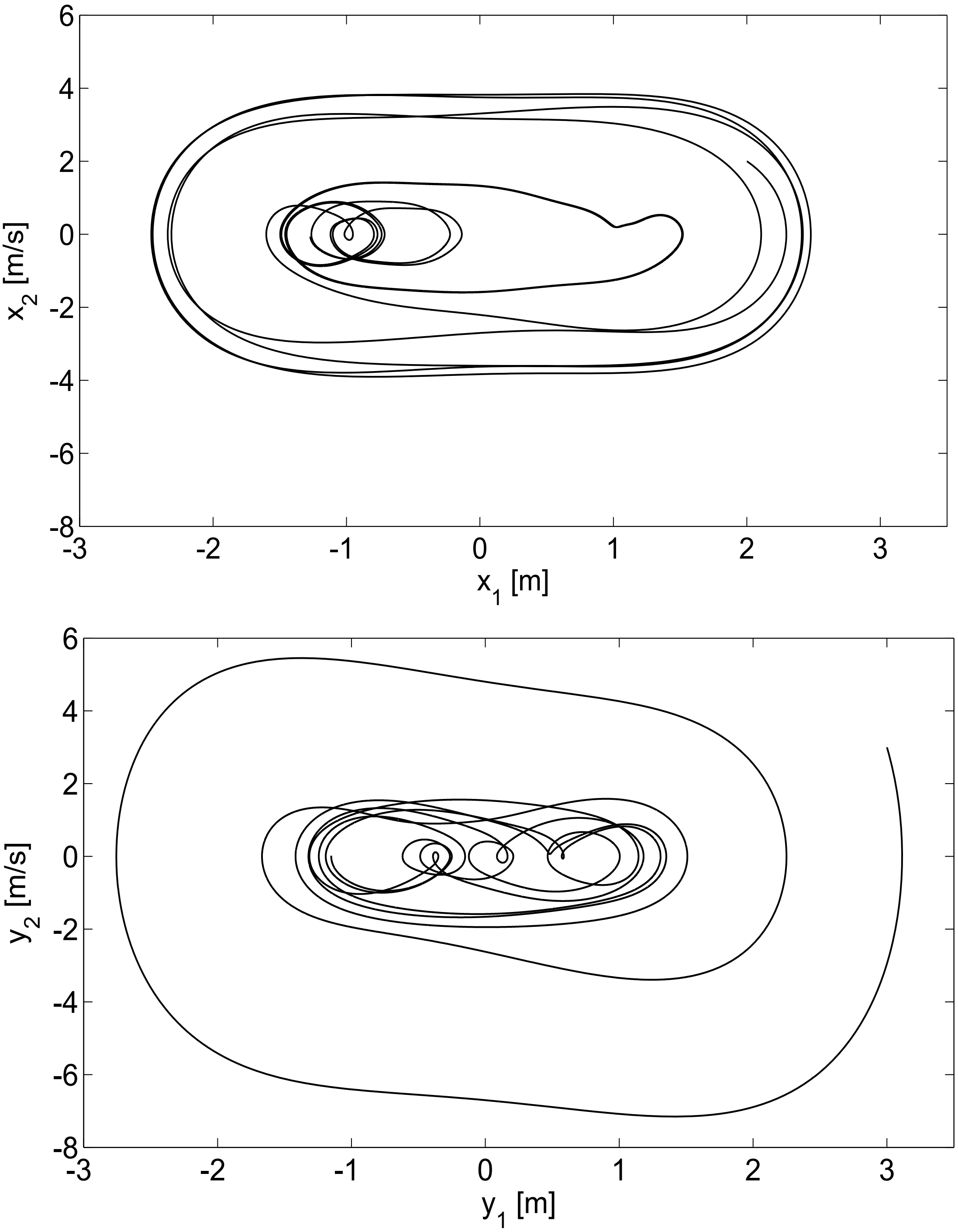


Figure 3.8: The chaotic motion of the master (upper) and slave (lower) Duffing systems.

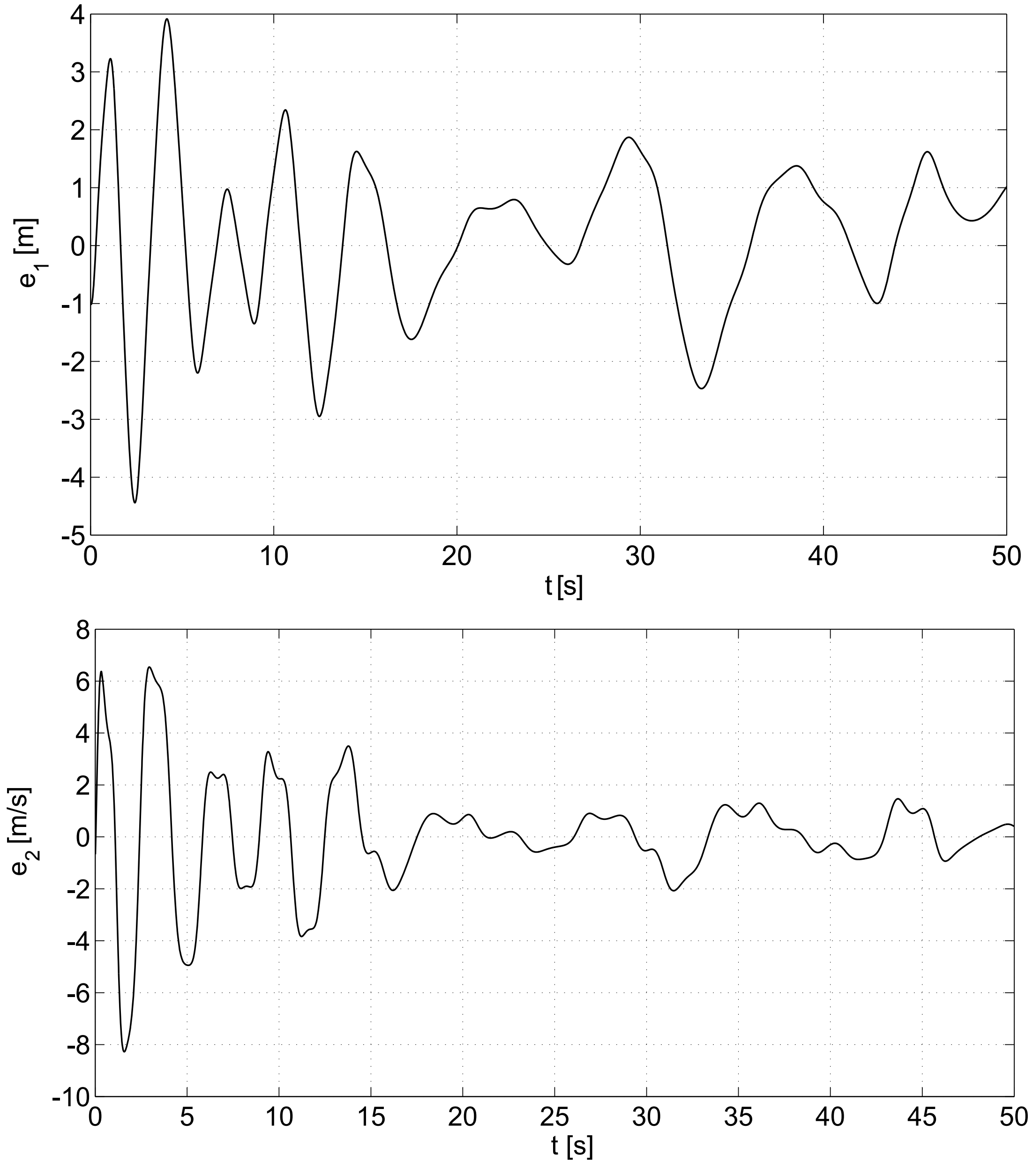


Figure 3.9: Tracking error of the first (upper) and second (lower) variables of the Duffing systems: $e_1 = x_1 - y_1$, $e_2 = x_2 - y_2$.

the types of Robust Fixed Point Transformations (introduced in Chapter 4), then the improvement achieved by the method can be doubled.

3.4 The model of the Φ^6-type Van der Pol oscillator

The equation of motion of the so-called Van der Pol oscillator was formulated in 1927 to model the behavior of an electrical circuit containing a triode valve [58]. Even today, this model is a popular paradigm at various scientific fields where the study of nonlinear oscillations and chaotic behavior has importance (see e.g. [59, 60, 61]).

The system can be described by the following equation, where x is the state variable:

$$m_{vdp}\ddot{x} - \mu_{vdp}\left(1 - x^2\right)\dot{x} + \omega_0^2 x + \alpha x^3 + \lambda_{vdp} x^5 = Q \tag{3.8}$$

in which m_{vdp} corresponds to some inertia, the term $-\mu\left(1 - x^2\right)\dot{x}$ symbolizes some nonlinear viscosity (i.e. dissipation for $|x| > 1$ and energy input for $|x| < 1$), ω_0^2 corresponds to some spring constant, while the remaining terms may describe further nonlinearities of this spring. The symbol Q here stands for some excitation force. The adaptive control's task is to exert proper force Q in order to keep the system's motion in the vicinity of a nominal trajectory.

In Chapter 8, it is shown that though the Robust Fixed Point Transformations method is effective, it can sometimes become unstable and start behaving like a Sliding Mode Controller, causing the so called "chattering effect". To avoid this disadvantage, an new algorithm is introduced by which the stability of RFPT can be regained in very short time. It is also proved and illustrated via simulation results (where the system is forced to follow a prescribed trajectory) that by applying the algorithm to the RFPT the tracking error of the Φ^6-type Van der Pol oscillator can be reduced significantly.

3.5 The cart-pendulum system

The cart-pendulum systems are one of the benchmark problems in the control area. They are often used to demonstrate methods in linear control such as the stabilization of unstable systems. Since these systems are nonlinear, they have also been useful in presenting some of the ideas in nonlinear control. The basic of the model is an inverted pendulum attached to a cart which is moved by a motor. The user is able to set the position and velocity of the cart through this motor and the track restricts the cart to move in one horizontal line. Sensors are attached to the cart and the pivots in order to measure the cart position and the joint angle of the pendulum, respectively. Figure 3.10 describes how the model is built.

The system's state propagation can be described by two equations where the following parameters (marked in Fig. 3.10) are used, respectively: M denotes the weight of the cart, while m and L_p are related to the weight and the length of the pendulum. x_c stands for the linear position of the cart and θ denotes the angular rotation of the hinge.

The Euler-Lagrange equations (taken from [62]) of motion describing this system and the applied torques (F on the cart and F_p on the pendulum) are

$$(M+m)\,\ddot{x}_c + b_{cp}\dot{x}_c + mL_p\ddot{\theta}\cos\theta - mL_p\dot{\theta}^2\sin\theta = F \tag{3.9}$$

$$mL_p\left(\cos\theta\ddot{x}_c + L_p\ddot{\theta} - g\sin\theta\right) = F_p = 0 \tag{3.10}$$

where b_{cp} parameter marks for the friction of the cart, while g stands for the gravity. Since the pendulum is not driven directly, F_p has to be 0. From (3.9) and (3.10) the second derivatives can easily be calculated:

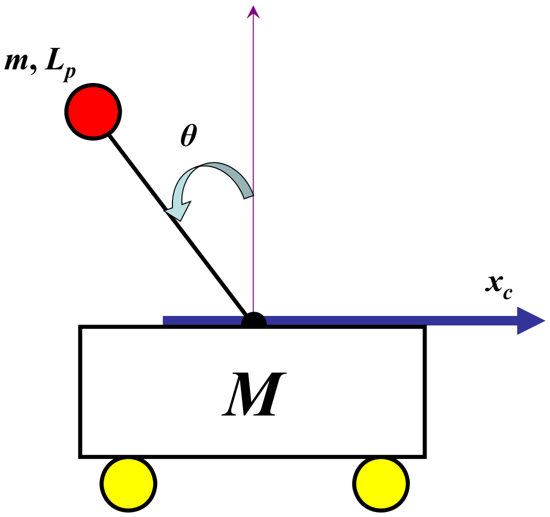


Figure 3.10: The cart-pendulum system got from [62].

$$\ddot{\theta} = \left((M+m)\, g \sin\theta + \mu_{cp} \cos\theta \right) / K L_p \tag{3.11}$$

$$\ddot{x}_c = -\left(m g \sin\theta \cos\theta + \mu_{cp} \right) / K \tag{3.12}$$

where

$$K = m \cos^2\theta - (M+m) \tag{3.13}$$

$$\mu_{cp} = F + m L_p \dot{\theta}^2 \sin\theta - b_{cp} \dot{x}_c \tag{3.14}$$

In Chapter 9, a new, Robust Fixed Point Transformations-based Fuzzy Logic Controller is presented for the same problem using the same values showing that RFPT can be combined with any Fuzzy Logic Controller in case an approximate model is used in the control task. As result the tracking error can significantly be reduced.

3.6 The dynamic model of the cart plus double pendulum system

The basic idea of the cart plus double pendulum system is similar as described in Section 3.5. There are two pendulums connected to the cart (instead of one), they are assembled on the cart by parallel shafts. Each pendulum has a rod of negligible mass, but considerable length: L_1 and L_2, and at the end of them there is a ball of negligible size, and of considerable mass: m_1 and m_2. The cart has a body of considerable mass: M, and wheels of negligible masses and momentums. The angular rotations of the hinges are denoted by θ_1 and θ_2, while the position of the cart is marked by x_c. The sketch of the cart plus double pendulum system is given in Fig. 3.11. The Euler-Lagrange equations of motion of this system are as follows:

$$\begin{aligned} &\begin{bmatrix} m_1 L_1^2 & 0 & -m_1 L_1 \sin\theta_1 \\ 0 & m_2 L_2^2 & -m_2 L_2 \sin\theta_2 \\ -m_1 L_1 \sin\theta_1 & -m_2 L_2 \sin\theta_2 & M + m_1 + m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{x}_c \end{bmatrix} + \\ &\quad + \begin{bmatrix} m_1 L_1 g \cos\theta_1 \\ m_2 L_2 g \cos\theta_2 \\ -m_1 L_1 \cos\theta_1 \dot{\theta}_1^2 - m_2 L_2 \cos\theta_2 \dot{\theta}_2^2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \end{aligned} \tag{3.15}$$

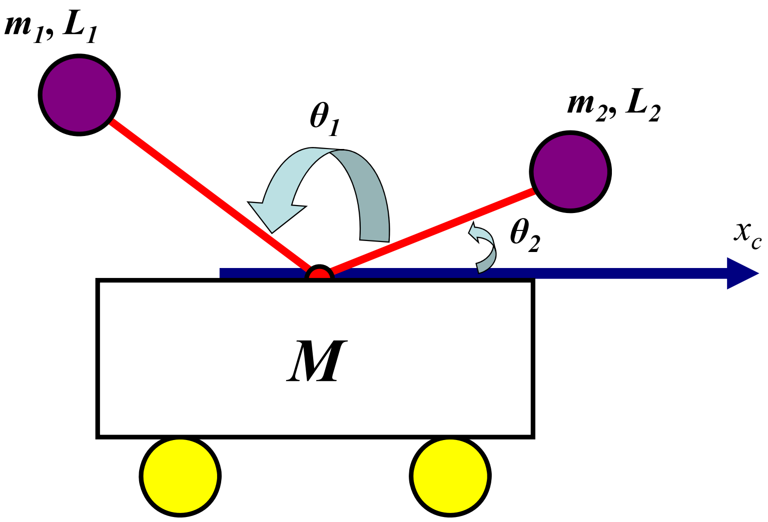


Figure 3.11: The cart plus double pendulum system.

in which the $F_3 \neq 0$ condition expresses the car's driving system. Evidently, if $\sin\theta_1 = \sin\theta_2 = 0$ simultaneously in (3.15) then neither $\ddot{\theta}_1$, nor $\ddot{\theta}_2$ give any contribution to F_3. Therefore, $\ddot{x}_c$ cannot be controlled by prescribing $\ddot{\theta}_1$ and $\ddot{\theta}_2$, which means that the underactuated system is dynamically singular in these points.

In Chapter 7, a new fuzzy-like parameter tuning of Robust Fixed Point Transformations is shown and investigated via simulations, where the cart plus double pendulum system is used. The parameter tuning gives an additional improvement achieved by RFPT.

3.7 Hydrodynamic models of freeway traffic

As mentioned earlier, Robust Fixed Point Transformations can be used advantageously when someone tries to examine a complex system approximated by a very rough model. One of the examples could be the emission rate of exhaust fumes of freeway traffic. The emission rate depends on a lot of things (e.g. road conditions, traffic properties, vehicle properties, meteorology, etc.), so exact modeling is a very difficult task. Thus instead of that the use of an approximate model can be suggested. If only two attributes (the vehicle density and the average speed of the cars) are considered and RFPT is applied then the incoming traffic can be controlled and by this the emission rate of exhaust fumes can be kept under a prescribed limit.

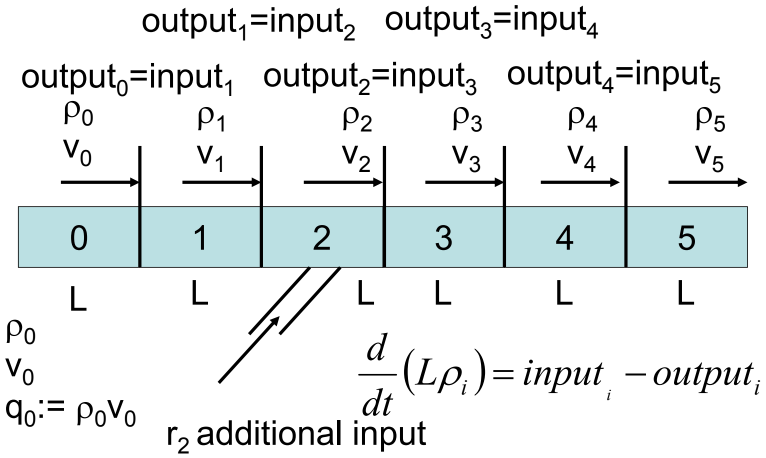


Figure 3.12: The discretized hydrodynamic model of freeway traffic, based on [63].

The approximate model is illustrated in Fig. 3.12. For simplicity, let us assume a one dimensional state variable for the road having six segments, from 0 to 5. The road segments have equal lengths denoted by L_{rs}. An additional side road (r_2) is applied at segment 2 where external vehicles can come in the traffic. Segments 0 and 5 represent the boundary conditions determining the propagation of the state variables of segments 1 to 4 by keeping time as continuous variable. A discretized approximation of the (in this case one dimensional) space variable is also applied. The state variables are the vehicle density (ρ), i.e. the number of vehicles over the road segment of unit length, and the velocity of the traffic (v). The quantity ρv $(1/s)$ denotes the traffic current density by the use of which “conservation” of the vehicles (they cannot disappear and new vehicles can appear only in road segment 0 or the side road) can be described by (3.16)-(3.19).

$$\dot{\rho}_1 = \frac{q_0 - \rho_1 v_1}{L_{rs}\lambda_{ft}} \tag{3.16}$$

$$\dot{\rho}_2 = \frac{\rho_1 v_1 - \rho_2 v_2 + r_2}{L_{rs}\lambda_{ft}} \tag{3.17}$$

$$\dot{\rho}_3 = \frac{\rho_2 v_2 - \rho_3 v_3}{L_{rs}\lambda_{ft}} \tag{3.18}$$

$$\dot{\rho}_4 = \frac{\rho_3 v_3 - \rho_4 v_4}{L_{rs}\lambda_{ft}} \tag{3.19}$$

$$\dot{v}_1 = \frac{V(\rho_1) - v_1}{\tau} + \frac{v_1(v_0 - v_2)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\rho_2 - \rho_0}{\rho_1 + \kappa} \tag{3.20}$$

$$\dot{v}_2 = \frac{V(\rho_2) - v_2}{\tau} + \frac{v_2(v_1 - v_3)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\rho_3 - \rho_1}{\rho_2 + \kappa} - \frac{\delta}{L_{rs}} \frac{r_2 v_2}{\rho_2 + \kappa} \tag{3.21}$$

$$\dot{v}_3 = \frac{V(\rho_3) - v_3}{\tau} + \frac{v_3(v_2 - v_4)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\rho_4 - \rho_2}{\rho_3 + \kappa} \tag{3.22}$$

$$\dot{v}_4 = \frac{V(\rho_4) - v_4}{\tau} + \frac{v_4(v_3 - v_5)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\rho_5 - \rho_3}{\rho_4 + \kappa} \tag{3.23}$$

where τ, η, κ, and δ are free parameters.

The dynamic behavior of this system is described by (3.19)-(3.23) in which for function $V(\rho)$ (which denotes the steady-state speed-density characteristic) various suggestions can be found in the literature as e.g. the Greenshields model [64]:

$$V(\rho) := v_{free} \left(1 - \frac{\rho}{2\rho_{cr}}\right) \tag{3.24}$$

and the Papageorgiou model [65]:

$$V(\rho) := v_{free} \exp\left(-\frac{1}{b}\left[\frac{\rho}{\rho_{cr}}\right]^b\right) \tag{3.25}$$

where ρ_{cr} denotes the critical vehicle density (density of jams), v_{free} marks the velocity of the traffic flow and b is a free parameter.

The first relationship established by Greenshields may provide negative velocities that are not allowed under normal conditions in a real traffic. The second one, the Papageorgiou model always results in interpretable nonnegative values. On this reason the latter approach is used.

If the additional ingress rate from the ramp (r_2) is applied at segment 2 then the appropriate dynamic model is given by (3.16)-(3.23). If electronically controlled road signs are used, the quantities ρ_0, v_0 (consequently $q_0 := \rho_0 v_0$), v_4 and v_5 can be set as constant boundary conditions which together with (3.23) with $\dot{v}_4 = 0$ immediately determine the density in segment 5 (ρ_5). Therefore the state variables remain only ρ_1, ρ_2, ρ_3, ρ_4, v_1, v_2, and v_3 for which we can define a coupled first order nonlinear differential equation system.

It must be noted that these equations describe a strongly "underactuated" system: there is only one time-varying control signal (r_2) that influences the propagation of seven state variables. Therefore there are three choices:

1. Only one of these variables is chosen and is precisely controlled.
2. An expression calculated from these variables is precisely controlled.
3. Some kind of optimal controller is applied. In this case, a cost function describes the weighted significance of the control of the individual variables.

In Chapter 11, after introducing the quasi-stationary solutions for the system, and making a stability analysis of these solutions a new controller is proposed for the quasi-stationary approach.

3.8 The qualitative properties of tire-road friction and the Burckhardt tire model

To model and analyze traffic-related phenomenons, in many of the cases it is necessary to model the vehicles themselves. Because of the increasing safety expectations of todays everyday people make tests of their cars (e.g. braking distance [66]). Many of the test results are public and can be used for comparative analysis for the determination of the effectiveness of an idea or method. This section focuses on braking problems since it not just helps to prevent crashes, but can save human life as well at e.g. difficult road conditions.

For designing braking systems, tire models are essential since they describe the connection between the road and the wheels. One of the most popular tire models was developed by Bakker, Pacejka and Lidner in 1989 [67]. This model obtained especial attention in the forthcoming 10 years (e.g. [68, 69, 70]). Its original form was developed in the form of a single–variable function that describes the dependence of the friction coefficient (μ) on the wheel slip: $\lambda_t := (v - r\omega)/v$, in which v denotes the velocity of the car body with respect to the road, r is the radius of the wheel, and ω describes the rotational velocity of the wheel axis. According to the sign selection convention in case $v > 0$ then the $v = r\omega$ condition corresponds to free rolling, $v < r\omega$ corresponds to skidding with accelerating, and the $v > r\omega$ condition pertains to skidding with braking. If motor brake is used in principle the $\omega < 0$ case also may occur. However, by the use of the conventional braking dials utilizing the friction forces, only the $0 \leq r\omega \leq v$ interval has physical meaning. The so-called "Magic Formula" developed by Bakker et al. uses an analytical formula for the function $\mu(\lambda_t)$ as

$$\mu(\lambda_t) = D \sin\left\{C \arctan\left[B\lambda_t - E(B\lambda_t - \arctan(B\lambda_t))\right]\right\} \tag{3.26}$$

that has geometrically well-interpreted parameters as “Stiffness Factor” (B), “Shape Factor” (C), “Peak Factor” (D), and “Curvature Factor” (E). Besides the fact that the identification of these parameters needs very sophisticated test equipment ([71]), it suffers from certain deficiencies from physical point of view: in the definition of λ_t if (3.26) is normalized according to v then the Magic Formula does not depend any more on $|v|$ (just the relative velocity of the skidding surfaces $v - r\omega$). To amend this “absence” Burckhardt suggested a modified version [72]:

$$\mu(\lambda_t, v) = e^{-B_4 v}\left[B_1\left(1 - e^{-B_2\lambda_t}\right) - B_3\lambda_t\right]. \tag{3.27}$$

For describing typical conditions, typical estimated values of these parameters are available. For describing typical road conditions Burckhardt determined appropriate values for parameters B_1, B_2, B_3, and B_4 [72], as shown in Table 3.1.

If the model is used for analysis of a braking process the term containing the coefficient B_1 increases with increasing λ_t while the term containing B_3 decreases. Therefore the maximum wheel slip is at some λ_{max} between 0 and 1. The coefficient with B_4 takes into consideration the dependence on $|v|$. The aim is to apply an appropriate braking strategy that keeps the deceleration of the wheel body at the prescribed value (if it is allowed by the road conditions) or keep it at the possible maximum. If braking starts at a free rolling state then at the beginning big negative $\dot{\omega}$ is needed until achieving the maximum of the friction coefficient. Thus, ω must be slowly decreased (as the velocity v also decreases) for keeping the friction coefficient near its maximum. A possible new strategy (with Robust Fixed Point Transformations) and a simple new vehicle model are suggested in Chapter 12, an the results are compared to public car test results.

Asphalt type	B_1	B_2	B_3	B_4
dry	1.2801	23.9900	0.5200	0.02
wet	0.8570	33.8220	0.3470	0.04
snowy	0.1946	94.129	0.0646	0.04

Table 3.1: Parameter setting for the different asphalt conditions got from [72].

3.9 Summary

In this chapter, several widely used nonlinear systems are summarized. First some chaotic attractors are detailed, like the FitzHugh-Nagumo neuron modell, the Duffing system, the Matsumoto-Chua circuit, and the Φ^6-type Van der Pol oscillator. Then, the cart-pendulum and the cart plus double pendulum systems are described. Finally, two realistic models are presented: the hydrodynamic model of freeway traffic, and the Burckhardt tire model. These systems are essential in understanding the new results in the next chapters.

4

Robust Fixed Point Transformations

In the previous chapter, those nonlinear systems are described that are essential for the analysis of the theses of this dissertation. In this chapter, the main process of the classical feedback control and the role of Robust Fixed Point Transformations are summarized. The following sections form the direct basis for the new results introduced in Chapters 5-11.

4.1 The expected-observed response scheme

Classical feedback control is a method that modifies the behavior of a system in a prescribed way. The expected-observed response scheme is the part of the classical feedback control. Usually, classical feedback control tasks are built as follows: There is a prescribed or "desired" behavior r^d for an existing system. The existing system has some kind of "excitation", for example some kind of torque or a control signal u which forces the system to produce the desired response. Different forces (gravity, friction, sometimes an accelerating motor, disturbance, etc.) take effect on the system. The actual value of the control signal has to be calculated with respect to these forces.

The control task can be formulated by the following equation

$$r^r = \varphi(u) \tag{4.1}$$

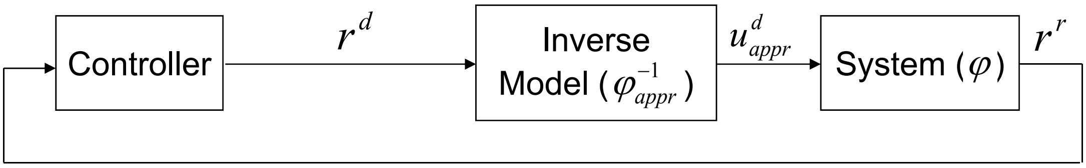


Figure 4.1: The block scheme of the classical feedback control Robust Fixed Point Transformations deal with.

which describes the correspondence of the control signal (u) and the actual system response r^r (after applying u on it). The main difficulty here is that the controlled system (with mapping φ) is not exactly known. For the proper control signal computation ($u^d = \varphi^{-1}(r^d)$) only approximate models (φ_{appr}^{-1}) can be of help:

$$u_{appr}^d = \varphi_{appr}^{-1}(r^d) \tag{4.2}$$

The approximation results in an error because the controller treats the approximate model as it was the desired one. The desired control force for the system could be achieved only by using an exact inverse model. So applying the approximate control signal to the system, gives the realized response. The correspondence between the realized (r^r) and the desired response (r^d) is

$$r^r \equiv \varphi(\varphi_{appr}^{-1}(r^d)) \equiv f(r^d) \neq r^d \tag{4.3}$$

The structure of the system can be seen in Fig. 4.1. Since the controlled system is not exactly known, neither can be determined function $f = \varphi_{appr}^{-1} \circ \varphi$. All that can be done is measure its output and based on it build a strategy to decrease the error.

4.2 The proof of the local convergence

As it is mentioned above it is hard to find the proper (desired) control signal u^d for an unknown system since the system's behavior cannot exactly be predicted. Using an approximate model may help to roughly determine the control signal (u_{appr}^d), but it is not always enough. Extra (usually iterative) calculations and time are needed.

The principle of RFPT is a strategy which shows how to determine an appropriate control signal which is able to control the not-exactly known system based on the approximate model. It constructs a function G_1 which transforms u^d_{appr} so that it gets closer to u^d: $|G_1(u^d_{appr}) - u^d| < |u^d_{appr} - u^d|$. In [21] authors show an iterative fixed point searching algorithm. Based on Banach's fixed point theorem [73] they prove that if

1. G_1 is differentiable and
2. $|G_1'| \leq M < 1$, where $M \in \mathbb{R}$

then

- The sequence constructed by G_1 is convergent and
- The fixed point of G_1 is the sequence's limit value.

In other words if the fixed point of function G_1 is u^d ($G_1(u^d) = u^d$), then the sequence built by G_1 converges to u^d. In the next few paragraphs a short proof of this statement is shown.

Take the next iterative sequence: $\{u_0,\ u_1 = G_1(u_0),\ \ldots\ ,\ u_{n+1} = G_1(u_n)\}$. If it is supposed that G_1 is differentiable then in the first step the contractivity of G_1 is shown. The definition of contractivity is $\exists\ 0 \leq M < 1\ \forall a, b \in \mathbb{R}\ |G_1(a) - G_1(b)| \leq M|a - b|$. If $|G_1'| \leq M < 1$, then

$$|G_1(a) - G_1(b)| = |\int\limits_a^b G_1'(u)\mathrm{d}u| \leq \int\limits_a^b |G_1'(u)|\mathrm{d}u \leq M|b - a| \tag{4.4}$$

If G_1 is contractive then a sequence built up with it is convergent in Banach spaces. Take the set of the real numbers $\mathbb{R}$ as a linear normed space where the two operations are the common addition and the multiplication with real numbers and the norm is the absolute value. In this case, this space is complete, therefore it is Banach space. So $\{u_n\}$ is convergent.

In the last step it is shown that the limit value of the sequence is the fixed point of G_1. If $L, n \in \mathbb{N}$ and $L, n > 0$ then

$$|u_{n+L} - u_n| = |G_1(u_{n+L-1}) - G_1(u_{n-1})| \leq \ldots \leq M^n|u_L - u_0| \to 0 \text{ as } n \to \infty \tag{4.5}$$

The proof contains a sequence of equivalent steps. It means that if the two conditions are fulfilled and the fixed point of G_1 is u^d, then the sequence constructed by G_1 will converge to u^d. So the next task is the construction of the proper mapping G_1. The next section deals with a possible representative.

4.3 The RFPT-based Model Reference Adaptive Controller

However, any appropriate G_1 function could be used, in the following, without limiting the universality, that function is shown and later applied which is suggested by Tar et.al. in [23]:

$$G_1\left(u, u^d_{appr}\right) = (u + K)\left(1 + B\tanh\left(A\left(h\left(u\right) - u^d_{appr}\right)\right)\right) - K \tag{4.6}$$

where $\varphi^{-1}_{appr}(\varphi(x)) = h(x)$, $h\left(u^d\right) = u^d_{appr}$, and

$$G'_1\left(u, u^d_{appr}\right) = \frac{(u + K)\, ABh'\left(u\right)}{\cosh^2\left(A\left(h\left(u\right) - u^d_{appr}\right)\right)} + 1 + B\tanh\left(A\left(h\left(u\right) - u^d_{appr}\right)\right) \tag{4.7}$$

In the equation A, B, and K are free parameters. They can be chosen so that the necessary limitation $\left|G'_1\left(u, u^d_{appr}\right)\right| < 1$ is guaranteed. It has two fixed points: u^d and $-K$. The latter can easily be excluded since there is more than one order of magnitude between the values of the two fixed points (further K is a free parameter chosen by the user). This means that if G_1 is flat enough around u^d, the iteration will converge to it, so $G_1(u, u^d_{appr})$ will get closer to u^d than u^d_{appr}. G_1 is robust with the respect to fluctuation of the system (with mapping φ). This robustness is a consequence of the strong nonlinear saturation of the sigmoid function tanh(), and can be approximately investigated by the use of an affine approximation of $\varphi(G_1(\varphi^{-1}_{appr}(x)))$ in the vicinity of u^d. The iteration generated by G_1 converge with a considerable speed even nearby u^d. Because of its robustness, function φ has less influence on its behavior.

Due to the principle of causality this iterative controller learns from the past experience therefore u and $h(u)$ can be calculated with one step delay. It means that they are got from the previous step. It causes one step delay in the computation and also a possible instability if the cycle time of the controller is “big”. If u^d varies quickly, thanks to the shift, u^d_{appr} can get out from interval where G_1 converges to u^d. In Chapter 7, it is introduced that this disadvantage can be fixed by a fuzzy-like parameter tuning.

If it is assumed that u^d varies slowly, then $G_1(u, u^d_{appr})$ is a proper choice. In practice the smaller A is, the wider "window" is got where function G_1 converges to u^d (instead of $-K$). After setting A, the better value is found for K, the quicker convergence is gained (B is always ±1, K is a very big negative and A is a very small positive number, they can be set by trial and error). Furthermore, tanh can be replaced by any bounded, strictly monotonously increasing differentiable $\sigma(x)$ function that fulfills the property $\sigma(0) = 0$, e.g. $\sigma(x) = x/(1 + |x|)$.

The controller's logic is very similar to that of the Model Reference Adaptive Controller (see Subsection 2.2.3): The controller (PD) determines the desired system response and then it is converted to the dimension of control forces so they together represent the reference model. The appropriate control force is calculated based on the variation of $h(u)$, so it is like the adaptation law. The RFPT part (G_1) calculates the proper control force, so it can be assumed as the controller. Finally the system is the system itself. The aforesaid RFPT-based Model Reference Adaptive Controller is shown in Fig. 4.2.

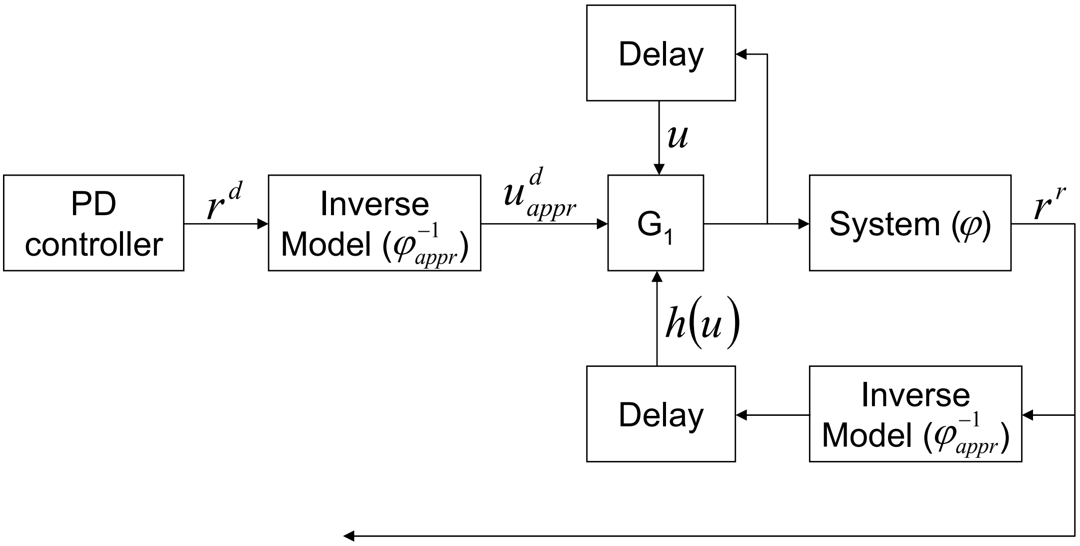


Figure 4.2: The block scheme of the Robust Fixed Point Transformations-based Model Reference Adaptive Controller.

4.4 The RFPT-based PD Controller

In the previous section, a method is shown how to find the proper control signal for a partly known system by using an approximate model. It is based on the idea of appropriately transforming the control signal calculated by the PID controller. In this section, a second approach is described for the problem.

The second option is to ameliorate the PID controller's result is finding the function G_2^d, which maps the previously explained desired state of the system r^d to some r_* so $\varphi_{appr}^{-1}(r_*) = u^d$ and $G_2^d(r^d) = r_*$. Since the difference between φ_{appr}^{-1} and φ^{-1} is not known, this is also not possible in practice. The other option can be to construct a function G_2 which at least takes r^d closer to r_*. In other words, function G_2 has to be applied to transform r^d by using the inverse approximation of model: $|G_2(r^d) - r_*| < |r^d - r_*|$.

The iterative fixed point searching algorithm, defined in Section 4.2 can be applied here, too. The proof of convergence is the same. The only thing that remains is the construction of the proper mapping G_2. In [21] the following function is proposed:

$$G_2\left(r, r^d\right) = (r + K)\left(1 + B\tanh\left(A\left(f\left(r\right) - r^d\right)\right)\right) - K \tag{4.8}$$

where $\varphi(\varphi_{appr}^{-1}(x)) = f(x)$, $f\left(r_*\right) = r^d$, and

$$G_2'\left(r, r^d\right) = \frac{(r + K)\,ABf'\left(r\right)}{\cosh^2\left(A\left(f\left(r\right) - r^d\right)\right)} + 1 + B\tanh\left(A\left(f\left(r\right) - r^d\right)\right) \tag{4.9}$$

This function has the same conditions as (4.6). It also has the one step delay, so $G_2(r, r^d)$ is a proper choice only if r^d varies slowly. The system with the RFPT-based PD Controller is shown in Fig. 4.3.

4.5 Summary

In this chapter, the basics of Robust Fixed Point Transformations are explained. First, the concept of the classical feedback control is detailed where RFPT can be used, then proof of RFPT's convergence is shown. Finally, two possibilities for RFPT-based classical controllers can be seen: the RFPT-based Model Reference Adaptive Controller and the RFPT-based PD controller. The most important difference between the two

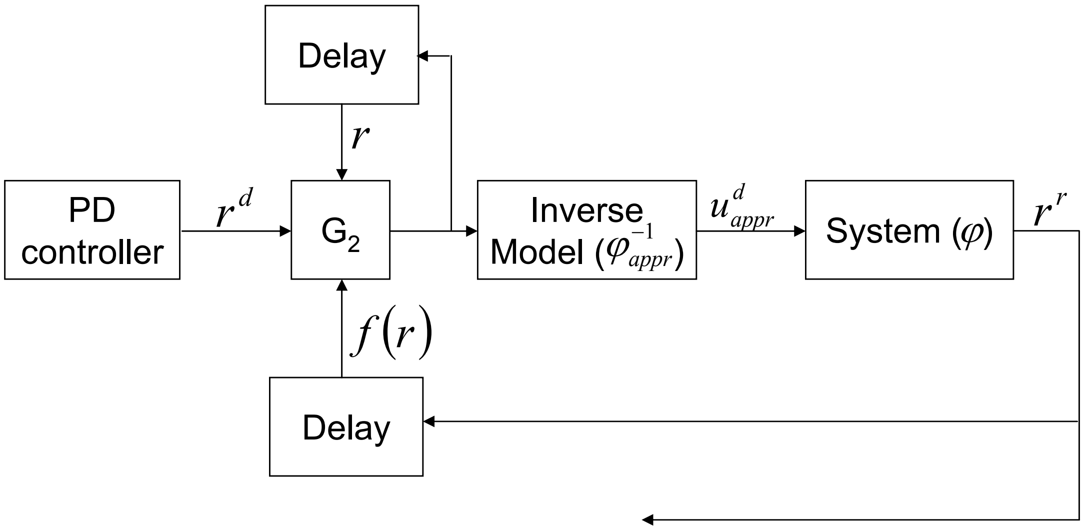


Figure 4.3: The block scheme of the Robust Fixed Point Transformations-based PID controller.

structures is that RFPT improves their results in different dimensions: in the dimension of the control forces and the state variables, respectively.

5

Robust Fixed Point Transformations in chaos synchronization

After providing a brief review of the litarature, in this chapter, the first new results are presented. A possible new application area of Robust Fixed Point Transformations is proposed and investigated: the field of chaos synchronization. Some chaotic oscillators are first modeled, then approximated, and finally controlled with RFPT-based controllers to prove the effectiveness of RFPT. In the later chapters similar approaches can be seen when new algorithms for RFPT are tested, and other RFPT-based controllers are designed for different chaotic systems.

5.1 Introduction

Nowadays, improving an existing system's results (speed, accuracy, efficiency, etc.) is very important since there is competition between manufacturers of machines, like industrial machines, cars, public vehicles, robots and labor-saving devices. On one hand, accuracy is more and more expected. On the other hand, avoiding a system to behave in an unwanted way is an other important issue.

Chaotic phenomena are present in our lives, especially in the nature. Usually, dynamical systems produce chaotic behavior. Unfortunately, chaos is present also in the control engineering and may cause major problems, like the sliding mode controllers' chattering effect or the more familiar phenomenon of car skidding. Generally, chaos is something that scientist want to stop.

A possible way to analyze, handle, or avoid chaotic behavior is chaos synchronization, this is why it is an important research issue. In chaos synchronization different attractors are typically used. For example, the Chua circuits [55] are used in electricity, Rössler's attractors [74] are applied in chemistry, the Lorenz systems [75] which were developed to model atmospheric convection and the Duffing systems [57] to model oscillators.

Regarding accuracy, there is an extra problem the manufacturers have to deal with. Replacing an existing system with a better one is sometimes very expensive so cheap modifications can bring more financial benefit. Scientists realize this monetary problem also: many papers can be found in the literature how to upgrade existing controllers' results without modifying it considerably.

One of the possible ameliorating methods is the so called Robust Fixed Point Transformations (RFPT) [23] which can improve existing controllers' results without too many modifications in the actual system. In this chapter, author shows that Robust Fixed Point Transformations can be advantageously utilized in a new field: the field of chaos synchronization. The chaotic systems presented in Chapter 3 are applied here, as well. Approximate models are constructed and used for the different attractors, and RFPT-based controllers are designed for the synchronization. The only constraint for the approximate models is that the sign of the differentiated composition of the system and the approximate model has to be known. After that the results are analyzed by simulations.

5.2 The synchronization of two FitzHugh-Nagumo neurons

In this section, the RFPT-based synchronization of two coupled FitzHugh-Nagumo neurons are presented. The background of the FitzHugh-Nagumo neuron models is presented in Section 3.1. First, a simple approximate model and a controller are constructed according to the requirement detailed above.

The approximate model for the control process is designed to have a similar structure to the original system (see Section 3.1). For the illustrative examples the approximate parameters are set to $\hat{a} = 12$, $\hat{b} = 1.5$, and $\hat{g}_c = 0.5$. For determining the desired state of the system a PD controller is chosen as $\ddot{x}_3^{Des} = \ddot{x}_1 + 2\Lambda(\dot{x}_1 - \dot{x}_3) + \Lambda^2(x_1 - x_3)$ with $\Lambda = 2/s$. For improving the controller the second RFPT-version is chosen (see Section 4.4).

In the second step the simulations are made in Scilab-5.1.1 [76] (developed by the Consortium Scilab (DIGITEO)) and the related graphical programming tool SCICOS 4.2. In the illustrated examples the free parameters of function G_2 are set to $B = 1$, $A = 5 \times 10^{-7}$ and $K = -10^6$. The applied parameter values (as written in Section 3.1) are $a_1 = 10$, $a_2 = 5$, $b_1 = 1$, $b_2 = 0.5$, $I(t) = A_{exct} sin(2\pi f_{exct} t)$ where $A_{exct} = 0.5$, and $f_{exct} = 1.271\, Hz$, $g_c = 0.02$. For the simulations the Scilab uses a solver for ordinary differential equations in which the integration method is automatically set by the system depending on the stiffness of the problem. The maximum step size of the integration is not limited, it is automatically set by SCICOS, too. The Integrator absolute tolerance parameter is set to 0.01, and the Integrator relative tolerance is 0.001. The cycle time for the delays is set to $\Delta t_{Cycle} = 2\, ms$. The initial states of the state variables are $x_1 = 0.005$, $x_2 = 0.005$, $y_1 = -0.005$, and $y_2 = -0.005$, a slight asymmetry is assumed. The disturbance forces are assumed to be high-frequency periodic signals with half amplitude of the external excitation I.

In the following illustrative simulation results are presented for comparing the traditional (C_1) and the RFPT-based PD controller (C_2). The disturbed control and the disturbed control are also compared. Figure 5.1 reveals the tracking error in the four cases. It can be seen that the error achieved by controller C_1 is reduced when RFPT is used. Figure 5.2 displays the realized responses vs. the desired ones. In ideal case one single line could be seen. With controller C_2 without disturbances the figure shows almost one single line, but with controller C_1 there is a significant difference between the system responses. With disturbances the system responses are chaotic, but controller C_2 achieves better results.

5.2.1 The effect of noise reduction on the synchronization two FitzHugh-Nagumo neurons

A very important area of research for nonlinear dynamical systems is the effect of noise, since noise can drastically modify the dynamics of a system. It can be present as a driving force or as an unwanted disturbance causing nondeterministic, or sometimes

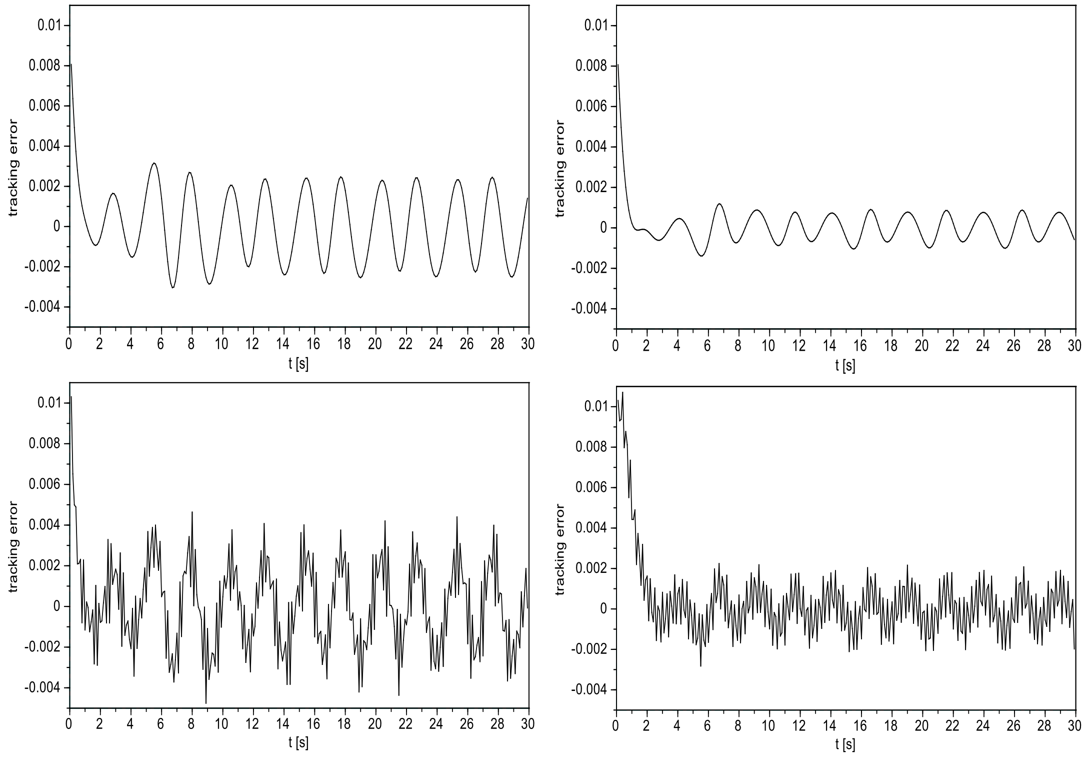


Figure 5.1: The tracking error $(x_1 - y_1)$ achieved by controllers C_1 and C_2: C_1 without disturbances (upper left); C_2 without disturbances (upper right); C_1 with disturbances (lower left); C_2 with disturbances (lower right). RFPT reduces the error every time to its one third.

chaotic behavior. The noise can appear as a consequence of a physical/dynamical attribute, e.g. thermal noise in every system warmer than absolute zero or the noise in nerve cells coming from synaptic events. In this case, basic physical knowledge is needed to understand, and if it is necessary, avoid the effects [77]. The other possibility is the unavoidable measurement noise, i.e. when the responses or state variables of a system are not calculated, but measured by some sensors affected by unknown external forces. When a sensor is used the user cannot assume that it measures everything correctly. Sensors always involve measuring noise. Luckily, nowadays many solutions exist to filter this kind of noise (e.g [78, 79, 80, 81, 82, 83, 84]). In the present literature the great majority of noise filters follow the model-based approach. The greatest impact on the field is given by Rudolf Kálmán who introduced the model-based state estimation in the early sixties [85, 86, 87, 88, 89].

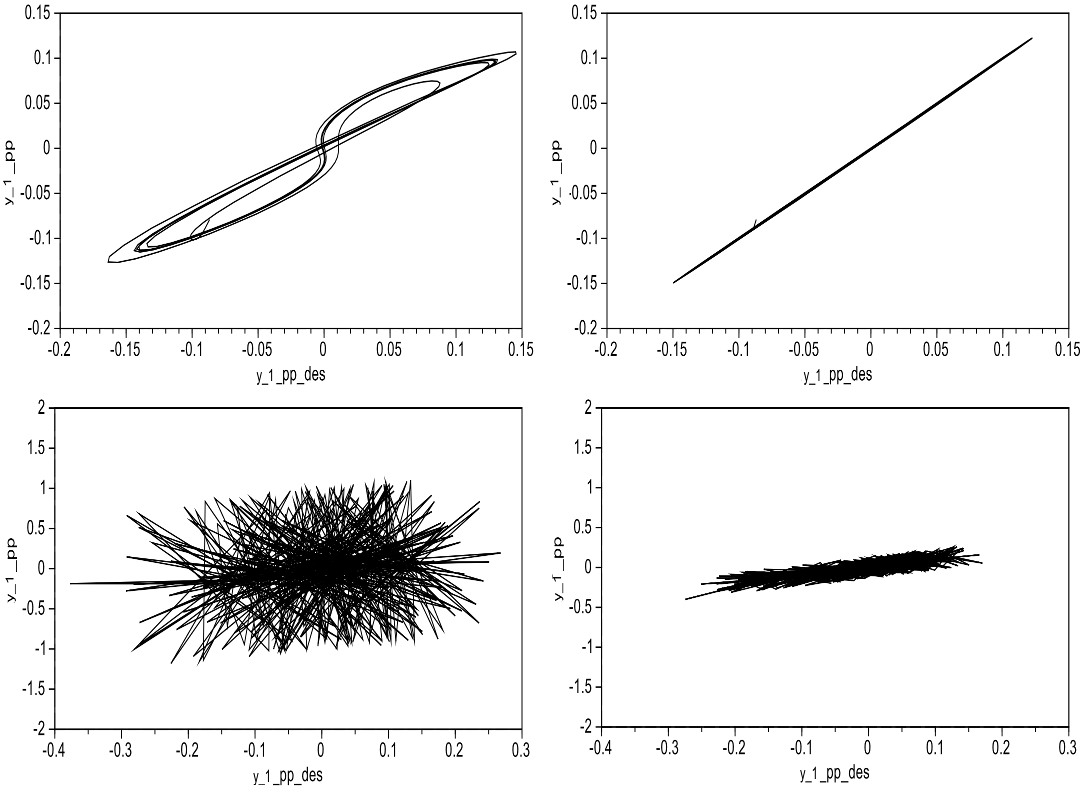


Figure 5.2: The realized $\ddot{y}_1$ values vs. the desired $\ddot{y}_1^{Des}$ values for controllers C_1 and C_2: C_1 without disturbances (upper left); C_2 without disturbances (upper right); C_1 with disturbances (lower left); C_2 with disturbances (lower right). In ideal case one single straight line could be seen. With controller C_2 the figure shows almost one straight line, but with controller C_1 there is a significant difference between the system responses.

The use of Kalman filters are based on certain assumptions regarding the statistical nature of the occuring noises. Furthermore, the Kalman filters have to be designed very carefully to avoid divergences [90].

A big disadvantage of Kalman filters is that whenever there is no reliable system approximation they cannot be applied. A different approach, the application of model-independent filters become necessary. In this section, it is examined how the filtering affects the performance of Robust Fixed Point Transformations. For the illustration, first a simple noise filter, an approximate model, and a controller are constructed.

A linear noise filter can be modeled in two ways in the time domain: in continuous case as an integral function as $\tilde{f}(t) := \int_0^\infty F(\tau)f(t-\tau)d\tau$ or in discrete case as a sum $\tilde{f}_k := \sum_{i=0}^{\infty} F_i f_{k-i}$. f denotes the systems' states, $F(\tau)$ is a monotonously decreasing function, and F_k-s are discrete weights that normally converge to zero as τ, $k \to \infty$. They correspond to the "forgetting speed" of the filter. They can be calculated in discrete case as $F_k := \beta^k(1-\beta)$ with $0 < \beta < 1$. For saving space and time F_k-s can be calculated by a simple buffer P: $P_{n+1} = \beta P_n + f_{k+1}$, $\tilde{f}_{k+1} = (1-\beta)P_{k+1}$. The actual value of β directly influences the "memory" of the filter: the larger value β has, the longer memory the filter has. In the following, the approximate model and the controller are detailed.

In fact, the approximate model is an inverse model as explained in Chapter 4. For simulation purposes, the same parameter settings are used as in the previous section: $\hat{a} = 12$, $\hat{b} = 1.5$, and $\hat{g} = 0.5$. By the use of the same PD controller relatively good tracking accuracy can be achieved if the feedback parameters are big enough. However, big feedback in the derivative term causes big disturbances in case of noisy signals. Within the frames of the PD controllers reduced feedback coefficients result in inaccurate tracking. The real virtues of the RFPT-based controller can be well observed when a relatively fast signal has to be slowly approximated. On this reason, in this section the same type of controller is applied with $\Lambda = 0.2 f_{exct}$. For improving the controller the second RFPT-version is chosen (see Section 4.4). In the following, simulation details and results are presented.

The task is to synchronize two chaotic FitzHugh-Nagumo neurons described in Section 3.1. In (3.2) x_1 and x_2 denote the master system while y_1 and y_2 describe the state of the slave system. Regarding the control task, assume that y_1 has to precisely track x_1 by properly setting $\dot{u}$. For calculating the differentiated control force, the observation of $\ddot{y}_1$ and $\ddot{x}_1$ is needed. Assume furthermore that there are simple sensors directly observing x_1 and y_1. The sensors have some observation noise that spreads through the numerical derivation with finite time-resolution Δt_{Cycle}. To reduce the effects of this noise the above detailed filter is used.

The simulations are made in Scilab-5.1.1 [76] (developed by the Consortium Scilab (DIGITEO)) and the related graphical programming tool SCICOS 4.2. For SCICOS the maximum step size of the integration is not limited in the simulations. In the illustrative examples the Integrator absolute tolerance parameter is set to 0.01, and the Integrator relative tolerance is 0.001. In the RFPT-based case the $\Delta t_{Cycle} = 0.1\,ms$ cycle time is chosen with $K = -1000000$, $A = 5 \times 10^{-7}$ and $B = 1$ settings. The initial values for the state variables are $x_1 = 0$, $x_2 = 0$, $y_1 = 0.005$, and $y_2 = 0$.

Since the adaptive controller works by observing $\ddot{x}_1$ and $\ddot{y}_1$ it may be sensitive to the observation noises. In this approach, in noisy case, it is assumed that the quantities x_1 and y_1 are directly observed as noisy signals. Their 1^{st} and 2^{nd} derivatives are estimated as $\dot{x}(t_n) \approx [x(t_n) - x(t_{n-1})]/\Delta t_{Cycle}$, $\ddot{x}(t_n) \approx [x(t_n) - 2x(t_{n-1}) + x(t_{n-2})]/\Delta t^2_{Cycle}$. The estimated values can be filtered accordingly. The $\beta = 0$ case corresponds to no filtering, while the $0 < \beta < 1$ corresponds to some filtering with shorter or longer filter memory.

The simulations are made to be able to compare two controllers: without RFPT (C_1) and with RFPT (C_2), to be able to compare results without and with measurement noise, and to be able to compare two noise filters: filter with $\beta = 0.9$ (F_1) and filter with $\beta = 0.5$ (F_2). In this section, six cases are presented: 1. with controller C_1, without noise, without filter; 2. with controller C_2, without noise, without filter; 3. with controller C_2, with noise, without filter; 4. with controller C_2, without noise, with filter F_1; 5. with controller C_2, with noise, with filter F_1; 6. with controller C_2, with noise, with filter F_2.

The first group of figures (Fig. 5.3) reveals the tracking error of the first state variable of the two systems $e = x_1 - x_3$ in all of the six cases. It reveals that without noise and filtering C_2 can gain more smooth and a little bit better trajectory tracking. With noise and without filter the initial error is very high (10000%) though later it converges to zero. The F_1 (without noise) does not increase the error, moreover it damps it a little bit. In the lowest figures, the results are shown with noise and with filters F_1 ($\beta = 0.9$) and F_2 ($\beta = 0.5$). F_1 gives significantly better results than the other filter.

The second group of figures (Fig. 5.4) reveals the connection of the desired and realized system responses $\ddot{y}_1^{Des} - \ddot{y}_1$. The bow–tie shaped figure in the first chart reveals that with C_1 (due to the parameter estimation errors) the trajectory tracking is not precisely realized. The effects of RFPT can be well observed in the second figure: $\ddot{y}_1$ becomes almost identical to $\ddot{y}_1^{Des}$. The last non chaotic figure (without noise but with filtering) does not differ from the second one which means that the filter does not "disturb" the improved results of RFPT. According to the simulations the noise filtering with $\beta = 0.9$ seems to work well. The other figures are noisy because the feedback in the derivative term of the PD controller causes disturbances in case of noisy signals.

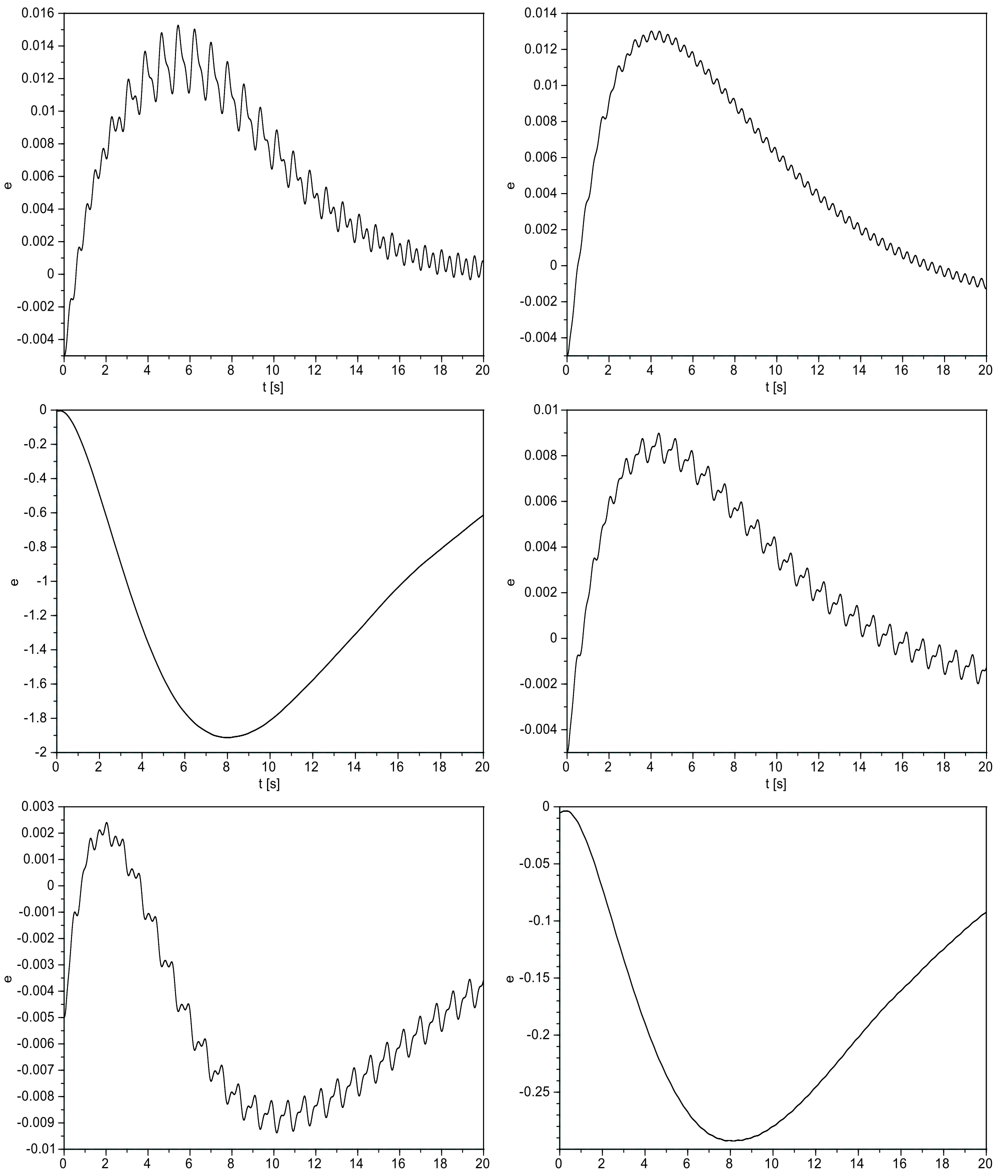


Figure 5.3: Tracking error of the first state variable $e_1 = x_1 - y_1$. Upper left: with controller C_1, without noise, without filter; Upper right: with controller C_2, without noise, without filter; Middle left: with controller C_2, with noise, without filter; Middle right: with controller C_2, without noise, with filter F_1; Lower left: with controller C_2, with noise, with filter F_1; Lower right: with controller C_2, with noise, with filter F_2. The filter improves the results in every case and does not disturb RFPT. $\beta = 0.9$ (F_1) seems to be the best choice.

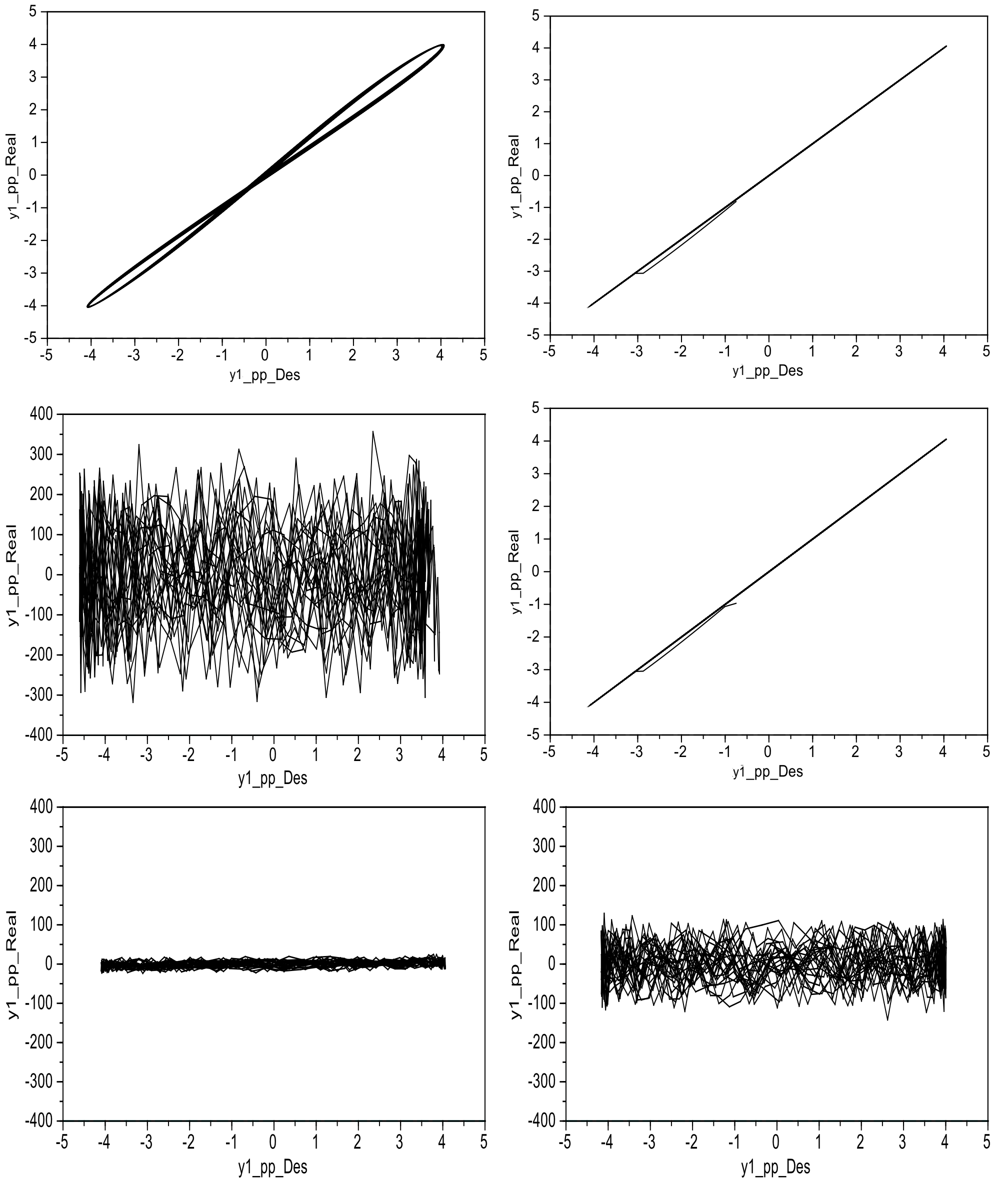


Figure 5.4: Response error $\ddot{y}_1^{Des} - \ddot{y}_1$. Upper left: with controller C_1, without noise, without filter; Upper right: with controller C_2, without noise, without filter; Middle left: with controller C_2, with noise, without filter; Middle right: with controller C_2, without noise, with filter F_1; Lower left: with controller C_2, with noise, with filter F_1; Lower right: with controller C_2, with noise, with filter F_2. Controller C_2 brings the best results. The other cases are noisy because the noise is not filtered out in this level. $\beta = 0.9$ (F_1) seems to be the best choice.

5.3 Synchronizing two Matsumoto-Chua circuits

In this section, the RFPT-based synchronization of two Matsumoto-Chua circuits is presented. The background of the circuits is presented in Section 3.2. First, a simple approximate model and a controller are constructed.

Without limiting the generality of the result, the following parameters are chosen for the illustrative examples: $C_1^{mc} = 1/10\,F$, $C_2^{mc} = 2\,F$, $L_c = 1/7\,H$, $G = 0.7\,\frac{1}{\Omega}$, $S_{small} = -0.1\,\frac{1}{\Omega}$, $S_{big} = -4\,\frac{1}{\Omega}$. The simulations are made in Scilab-5.1.1 [76] (developed by the Consortium Scilab (DIGITEO)) and the related graphical programming tool SCICOS 4.2. The maximum step size of the solver of SCICOS is identical with the cycle time of RFPT: $10^{-2}\,s$. The Integrator absolute tolerance parameter is set to 0.001, and the Integrator relative tolerance is 0.0001. The parameters values for function G_2 are $B = 1$, $K = -10000$, and $A = 10^{-4}$. In the simulations the slave system has the same initial conditions as the master system: $v_{C10} = 1.45305\,V$, $v_{C20} = -4.36956\,V$, and $i_{L0} = -0.15034\,A$.

The approximate model of the slave system is designed to have the same structure and parameter values than the master Matsumoto-Chua circuits: $\hat{C}_1^{mc} = C_1^{mc}$, $\hat{C}_2^{mc} = C_2^{mc}$, $\hat{L}_c = L$, $\hat{G} = G$, $\hat{S}_{small} = -0.05$, $\hat{S}_{big} = -3.5$. The slave system parameters are set differently: $\tilde{C}_1^{mc} = 0.9C_1^{mc}$, $\tilde{C}_2^{mc} = 0.8C_2^{mc}$, $\tilde{L}_c = 0.7L$, $\tilde{G} = 0.9G$, $\tilde{S}_{small} = -0.05$, $\tilde{S}_{big} = -3.5$. For determining the desired state of the system a PI controller is chosen $\dot{v}_{C2}^{Des} = \dot{v}_{C2}^{Nom} + \Lambda\left(v_{C2}^{Nom} - v_{C2}\right)$ with a small feedback gain $\Lambda = 2/s$. For improving the controller the second RFPT-version is used (see Section 4.4). In the following, simulation details and results are presented.

Hereunder, illustrative simulation results can be seen with PI control (C_1), and with RFPT-based PI control (C_2). In Fig. 5.5 the driven signals versus the reference signals can be followed. Figure 5.6 shows the tracking error versus time. The figures well reveal the improvement achieved by RFPT: with controller C_1 the system responses differ from each other significantly, but with controller C_2 the differences and the tracking error are reduced considerably. In Fig. 5.7 the control currents of the two controllers can be compared.

According to the simulations the performance of controller C_2 is far better than that of controller C_1 with small exponent $\Lambda = 2/s$. If Λ is increased then better results can be gained, but without the extension of function G_2 it does not reach the results of the RFPT-based controller with small Λ. Figure 5.8 shows the results of controller C_1 with $\Lambda = 10/s$.

5.4 Summary

In this chapter, a new application area of Robust Fixed Point Transformations is proposed and investigated: the field of chaos synchronization. Chaos synchronization is very useful to supervise natural processes and to test the effectiveness of controllers. For this purpose approximate models are built for different chaotic attractors and Robust Fixed Point Transformations-based controllers are designed to synchronize the chaotic systems based on the approximate models. Then the effectiveness of the proposed tools are analyzed by simulations. The results prove that the performance of the original controllers can significantly be increased with RFPT, and that even a poorly adjusted controller combined with RFPT outperforms a well set controller.

The results considered to be new have been published in journal paper [J2] and conference papers [C1, C4, C5, C6, C7, C8, C19]. Similar achievements with different attractors can be seen in Chapters 6 where two Duffing systems are synchronized, and in Chapter 8 where the chaotic Φ^6-type Van der Pol oscillator is controlled.

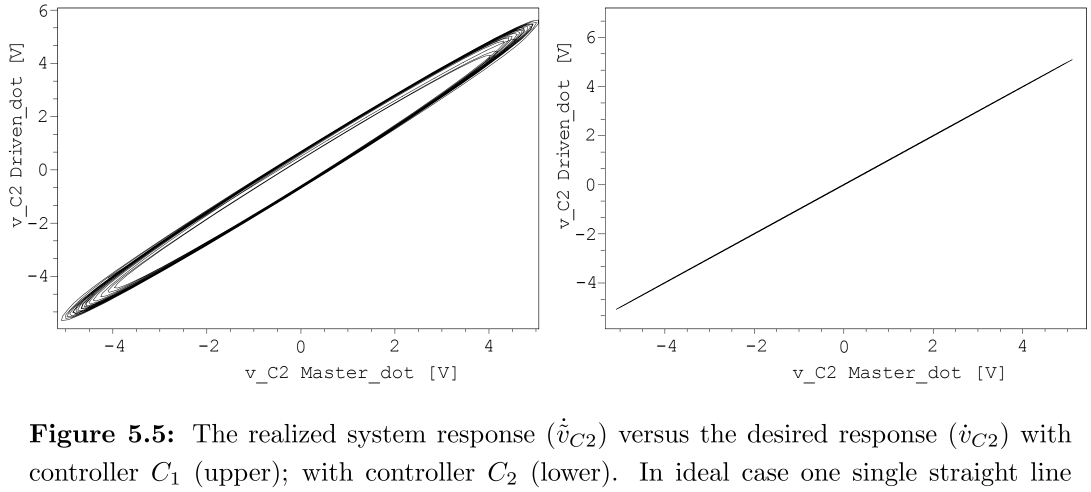


Figure 5.5: The realized system response ($\dot{\tilde{v}}_{C2}$) versus the desired response ($\dot{v}_{C2}$) with controller C_1 (upper); with controller C_2 (lower). In ideal case one single straight line could be seen. With controller C_2 the figure shows almost one straight line, but with C_1 there is a significant difference between the system responses.

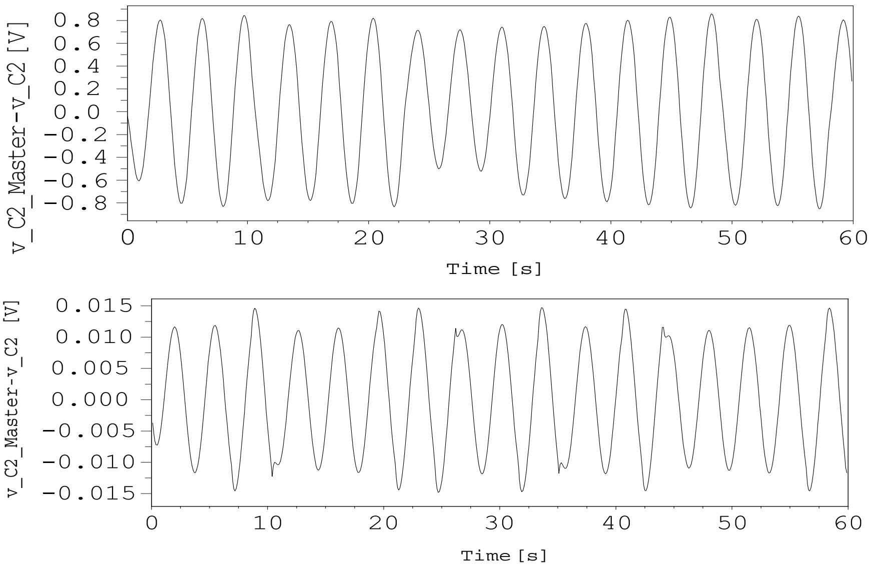


Figure 5.6: Tracking error $v_{C2} - \tilde{v}_{C2}$ with controller C_1 (upper); with controller C_2 (lower). The tracking error is reduced significantly with RFPT.

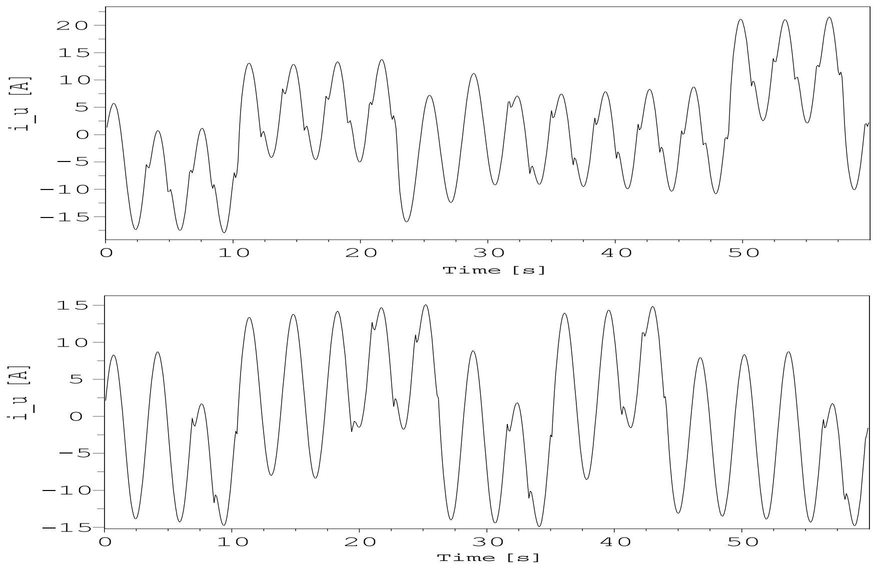


Figure 5.7: The control current versus time (in s units) with controller C_1 (upper); with controller C_2 (lower).

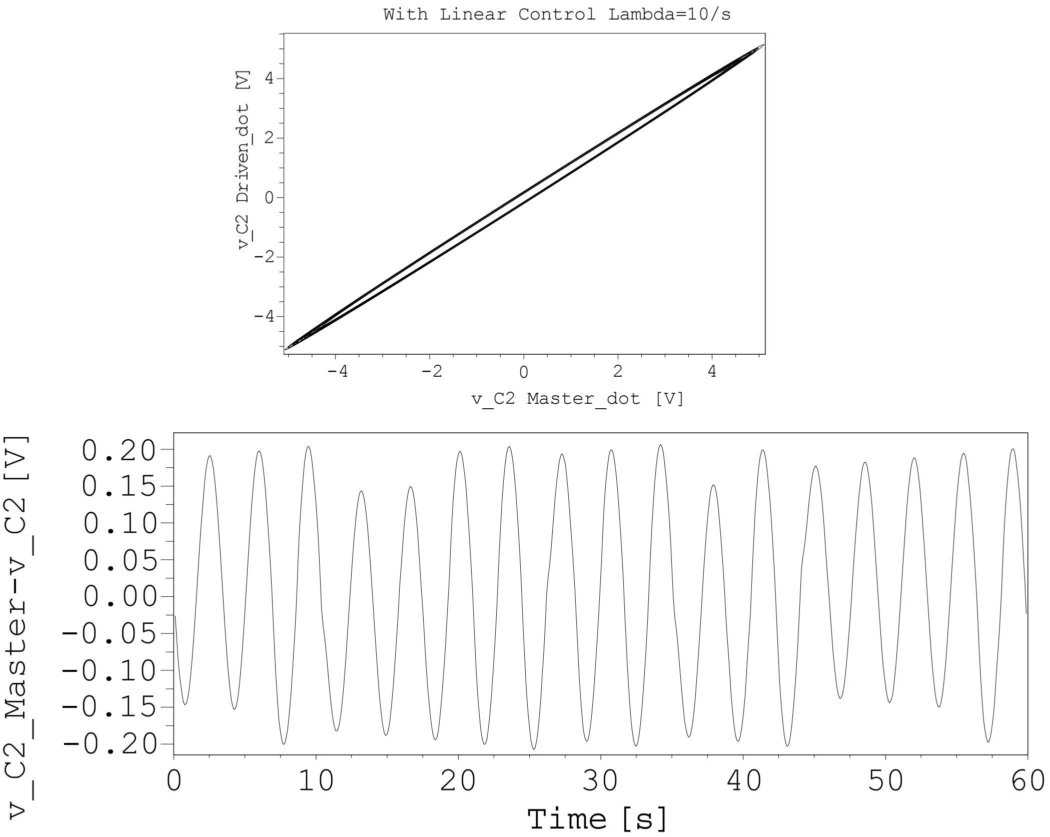


Figure 5.8: The results of controller C_1 with increased exponent $\Lambda = 10/s$: realized system response versus the desired response (upper); tracking error (lower). Similar performance to controller C_2 with small exponent cannot be achieved.

6

The “recalculated” Robust Fixed Point Transformations

In the previous chapter, a new application area for Robust Fixed Point Transformations is proposed and investigated: the field of chaos synchronization. In this chapter, based on the preliminary knowledge of RFPT, a new structure with an additional controller is introduced to improve existing controllers' results. The extra controller gains additional tracking error reduction compared to the original two versions. In the following, it is investigated if RFPT is able to improve the results of the soft-computing-based controllers.

6.1 The RFPT-based “recalculated” PD Controller

In Chapter 4 two options are reviewed how to build in a simple transformation (G) into a controller so that the system gives more accurate response. In the following, another possibility is shown how the system's results can be improved. The idea is based on the theory that in a simple feedback control there are three main tools: a controller ($PD()$), an approximate inverse model ($\varphi^{-1}_{appr}()$), and the system itself ($\varphi()$). In Chapter 4 it is shown how authors build in the improver function between the model and the system

$$\varphi(G_1(\varphi^{-1}_{appr}(PD(r^r_n)))) = r^r_{n+1} \tag{6.1}$$

and between the controller and the model

$$\varphi(\varphi_{appr}^{-1}(G_2(PD(r_n^r)))) = r_{n+1}^r \tag{6.2}$$

In this chapter, a new adaptation to Robust Fixed Point Transformations is introduced: a function further improving the performance is included between the system and the controller:

$$\varphi(\varphi_{appr}^{-1}(PD(G_3(r_n^r)))) = r_{n+1}^r \tag{6.3}$$

Based on the previous chapter's logic the goal is to find the function (G_3^d), which maps r^d to some r_G^* so that $PD\left(r_G^*\right)=r_*$ (where $\varphi_{appr}^{-1}\left(r_*\right)=u^d$ and $PD()$ denotes the PD controller). In effect this means that the controller has to be tricked about where the proper place for the system is. So it is forced to map its inputs to somewhere else. Since φ is still not known, the exact value of r_G^* still cannot be determined. All that can be done is constructing function G_3 which at least takes r^d closer to r_G^*, so $|G_3(r^d) - r_G^*| < |r^d - r_G^*|$.

The iterative fixed point searching algorithm is adaptable again. The same proof is valid in this case. The only thing that has to be done is the construction of the proper mapping G_3. Based on the previous results the following function is proposed:

$$G_3\left(PD(r), r^d\right) = (PD(r) + K)\left(1 + B\tanh\left(A\left(f\left(PD(r)\right) - r^d\right)\right)\right) - K \tag{6.4}$$

where $\varphi(\varphi_{appr}^{-1}(x)) = f(x)$, $f\left(PD(r_*^G)\right) = r^d$, and

$$G_3'\left(r, r^d\right) = \frac{(r+K)\,ABf'(r)}{\cosh^2\left(A\left(f(r) - r^d\right)\right)} + 1 + B\tanh\left(A\left(f(r) - r^d\right)\right) \tag{6.5}$$

This function has the same conditions as (4.6). It also has the shift inside, so $G_3(PD(r), r^d)$ is a proper choice only if r^d varies slowly.

The system with the RFPT-based "recalculated" PD Controller is shown in Fig. 6.1. The block scheme is very similar to the RFPT-based simple PD controller's (Fig. 4.3), but there is a significant difference: r^d is not just transformed, but a new desired response is calculated with the help of a second controller. This new desired response is calculated from the transformed desired response (transformed by G_3), so the inverse approximation is taken into account. The recalculation of the desired response is made by a second controller, which (in most cases) reduces additionally the tracking error. In the simulations the same controllers can be used, but ad-libitum, they can be different.

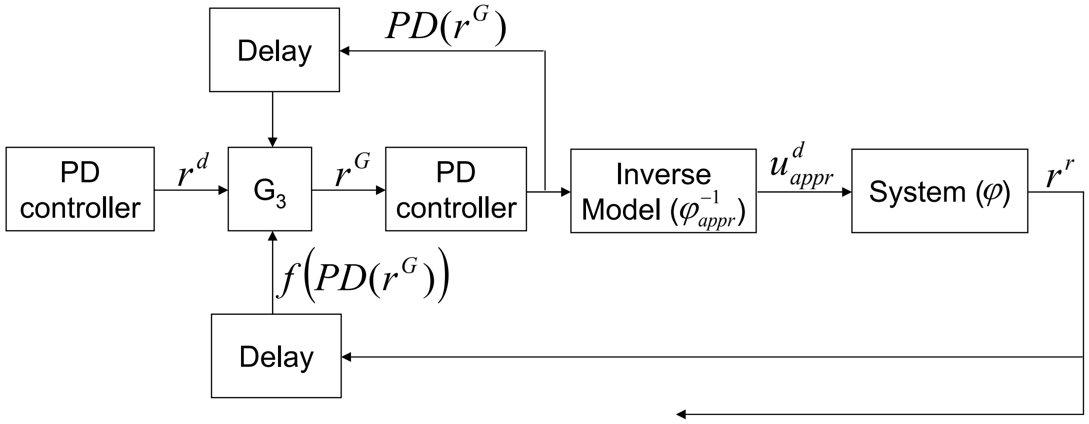


Figure 6.1: The block diagram of the RFPT-based "recalculated" controller scheme. An extra controller is added which causes the further decrease in the tracking error.

6.2 Simulation results

In this section, illustrative simulation results are presented using chaotic systems. The task is to synchronize two nonlinear Duffing systems that are not identical. The simulations are made by the MATLAB-Simulink pair. The programs use a solver for ordinary differential equations (ode45 [91]). The tracking error is strongly related to the integrator absolute tolerance. The tolerance is set organically in every case (10^{-3}) to be able to compare them. The maximum step size of the integration is also the same (10^{-3}) in every simulation. Without limiting the generality of the new result, in the examples the parameter values for the two Duffing systems are set to $\alpha_1 = 1N$, $\alpha_2 = 0.8N$, $\beta_1 = 1\frac{N}{m^2}$, $\beta_2 = 1.5\frac{N}{m^2}$, $\delta_1 = 0.2N$, $\delta_2 = 0.3N$, $\omega_d = 2Hz$, and $a = 1.2Nm$. The approximate inverse model has the same structure than the systems (see (3.6) and (3.7)), but different parameters are assumed: $\hat{\alpha} = 1.5N$, $\hat{\beta} = 0.5\frac{N}{m^2}$, and $\hat{\delta} = 0.1N$. For the tracking error reduction the following PD controller is used:

$$\dot{y}_2^{Des} = \dot{y}_2 + 2\Lambda(x_2 - y_2) + \Lambda^2(x_1 - y_1) \tag{6.6}$$

where $\Lambda = 5/s$.

The simulations are made both with and without disturbance. The tanh() function makes the RFPT-based controllers very robust, so the disturbances do not increase the tracking errors' order of magnitude in the RFPT-based simulations. That is why it is

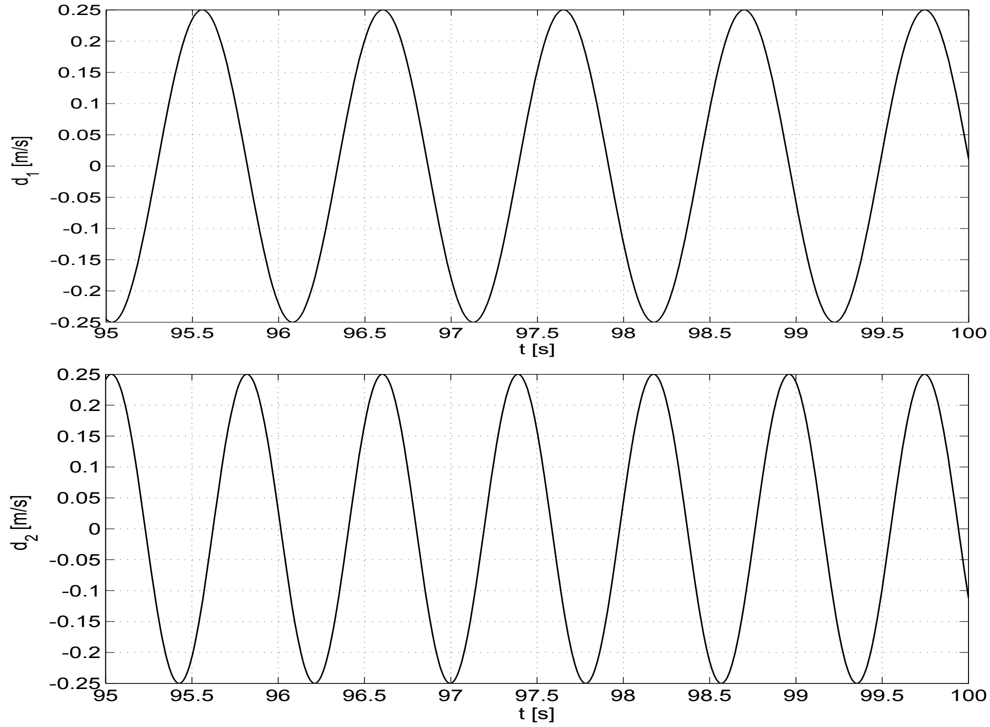


Figure 6.2: The disturbance forces applied on the master (upper) and the slave (lower) systems (defined in (3.6)-(3.7)).

not necessary to show the "disturbed" results. The disturbing sine waves are presented in Fig. 10.2.

The simulations are made by using PD controllers, in first case without (controller C_1 denotes double PD, C_2 marks MRAC, and C_3 stands for a simple PD controller), then based with all the three types of RFPT (the first two types are explained in Chapter 4; C_4 denotes the RFPT-based "recalculated" controller, C_5 marks the RFPT-based MRAC, and C_6 stands for the RFPT-based PD controller).

In the first step simulation results of the three different controllers are shown without the RFPT extensions (C_1, C_2, and C_3). Figures 6.3 and 6.4 illustrate the tracking errors of the first ($e_1 = x_1 - y_1$) and second ($e_2 = x_2 - y_2$) state variables. The figures reveal that synchronization cannot be achieved by simply using a Model Reference Adaptive Controller with model approximation (C_2) described in Section 4.3. The other two controllers (C_1 nad C_3) proved to be successful, but the proposed new "recalculated" structure C_1 gives more accurate results than C_3 (thanks to the second controller).

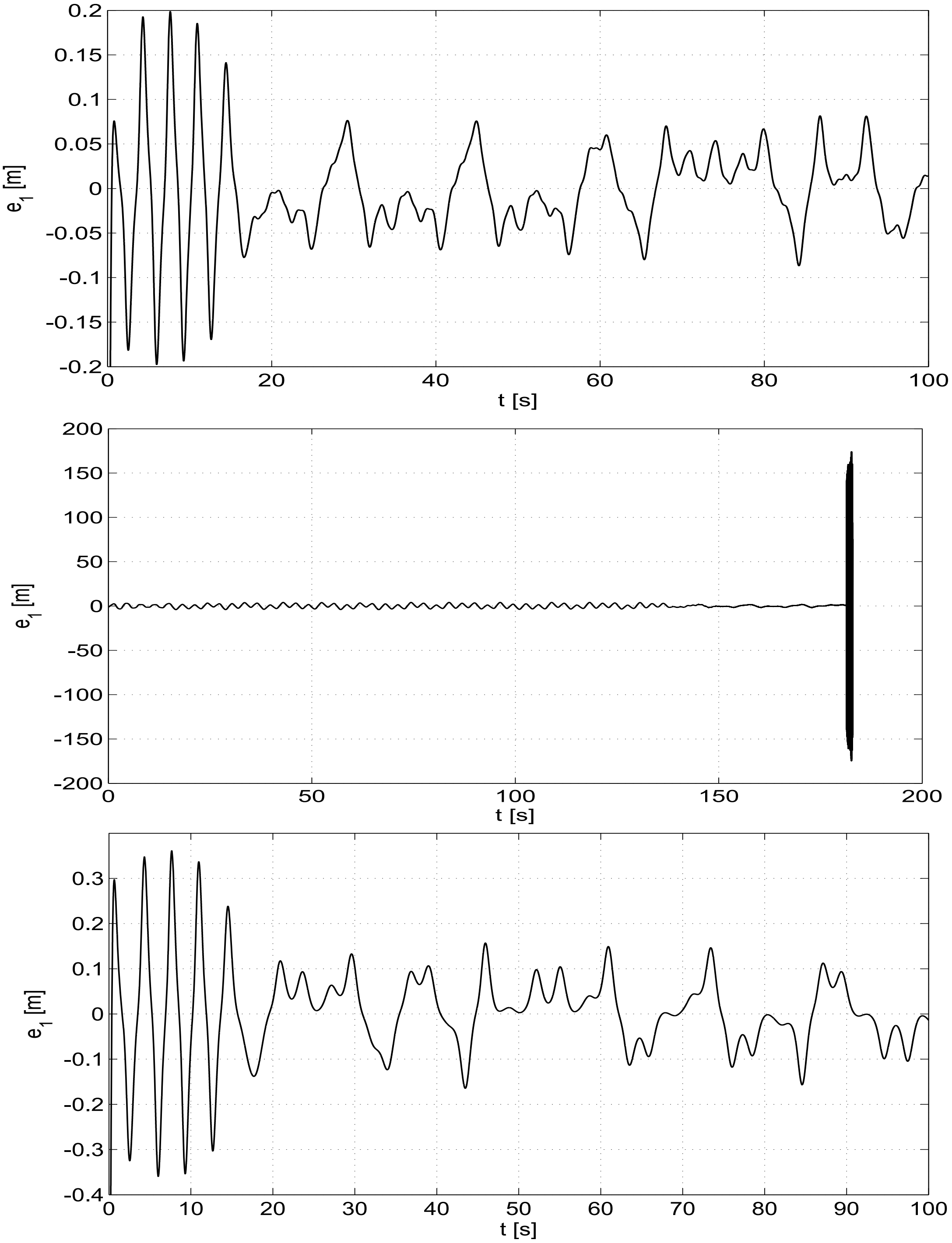


Figure 6.3: The tracking errors of the first state variable of the slave systems with controllers C_1, C_2, and C_3 ($e_1 = x_1 - y_1$); upper: C_1, middle: C_2, lower: C_3. As it is shown, C_2 may fail. C_3 might succeed, but does not accomplish as well as C_1 which has two PD controllers.

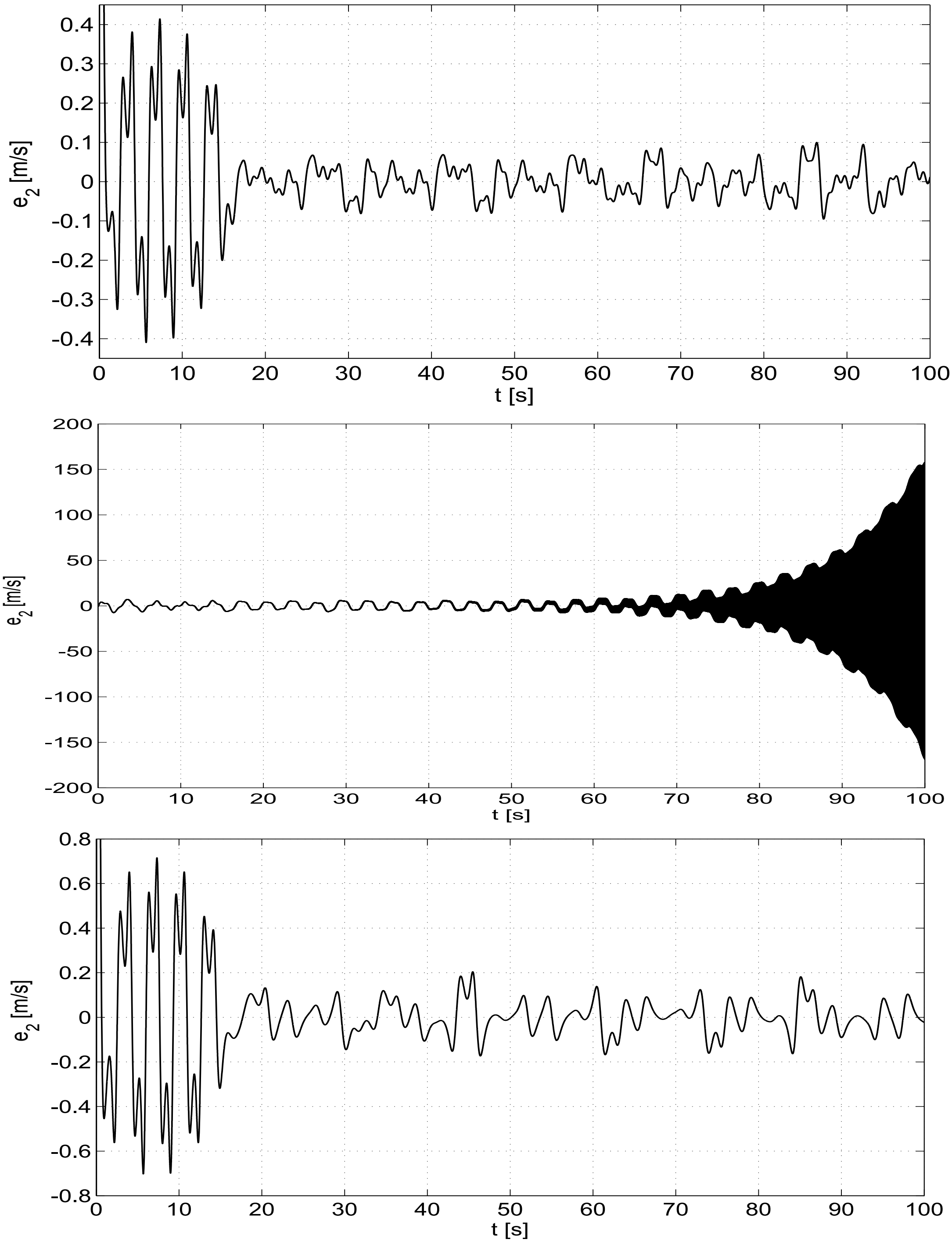


Figure 6.4: The tracking errors of the second state variables of the slave systems with controllers C_1, C_2, and C_3 ($e_2 = x_2 - y_2$); upper: C_1, middle: C_2, lower: C_3. As it is shown, C_2 may fail. C_3 might succeed, but does not accomplish as well as C_1 which has two PD controllers.

The differences between the desired and realized responses without RFPT are illustrated in Fig. 6.5. The MRAC version predicts the failure of the synchronization at the early stage of the simulation when the tracking errors are "small" yet. The "recalculated" structure achieves the most accurate results again.

In the last step simulation results are shown when applying RFPT. The values of the free parameters of G_1, G_2, and G_3 are marked in Table 6.1. Figures 6.6 and 6.7 illustrate the tracking errors of the first and second state variables. The figures reveal that the RFPT-based traditional PD controller (C_6) lowers the tracking error by more than two orders of magnitude. The RFPT-based MRAC (C_5) now succeeds and generates similar tracking error than C_6. The proposed "recalculated" controller (C_4) lowers the error with an additional 50% compared to the other methods.

The differences between the desired and realized responses in the RFPT-based case can be seen in Fig. 6.8. C_5 and C_6 result in similar errors. The proposed "recalculated" controller C_4 bisects the error here, too.

6.3 Summary

Robust Fixed Point Transformation is often applied to improve existing and well behaving controllers' results in case an approximate model is used in the control task. In this chapter, a new approach to Robust Fixed Point Transformations is introduced. The approach is based on the idea of integrating a second controller into the system. The great advantage of the second controller is that with the help of it the proposed new RFPT-based "recalculated" controller considerably decreases the tracking error

	A	B	K
G_1	2×10^{-5}	-1	70000
G_2	10^{-2}	1	-100
G_3	10^{-2}	1	-100

Table 6.1: The values of the free parameters of G_1, G_2, and G_3.

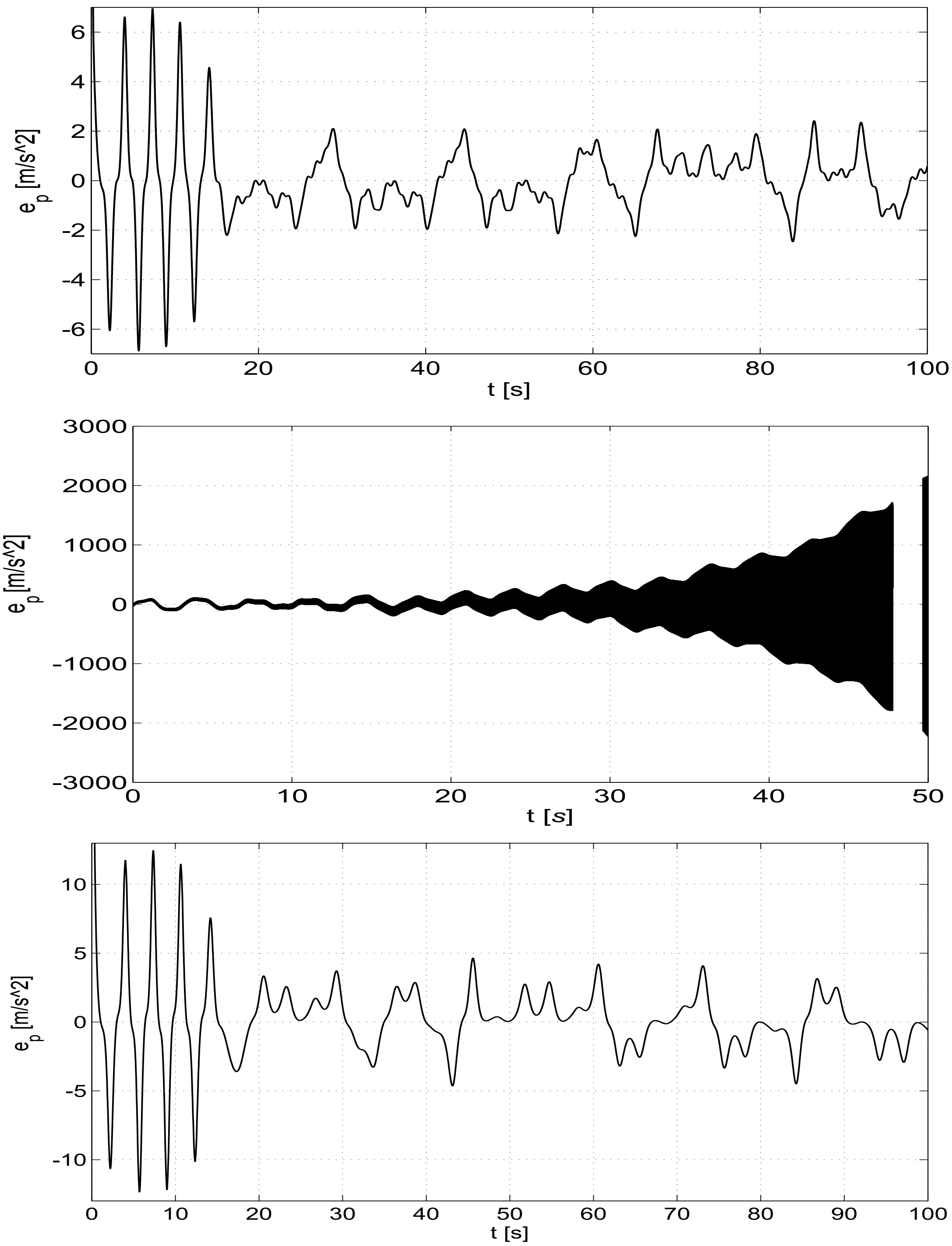


Figure 6.5: The difference between the desired and the realized response ($e_p = \dot{y}_2^d - \dot{y}_2^r$) with controllers C_1, C_2, and C_3 ($e_2 = x_2 - y_2$); upper: C_1, middle: C_2 (0-50 seconds), lower: C_3. As it can be seen, C_2 may predict the failure in early stage. C_3 might succeed, but not as well as C_1 which has two PD controllers.

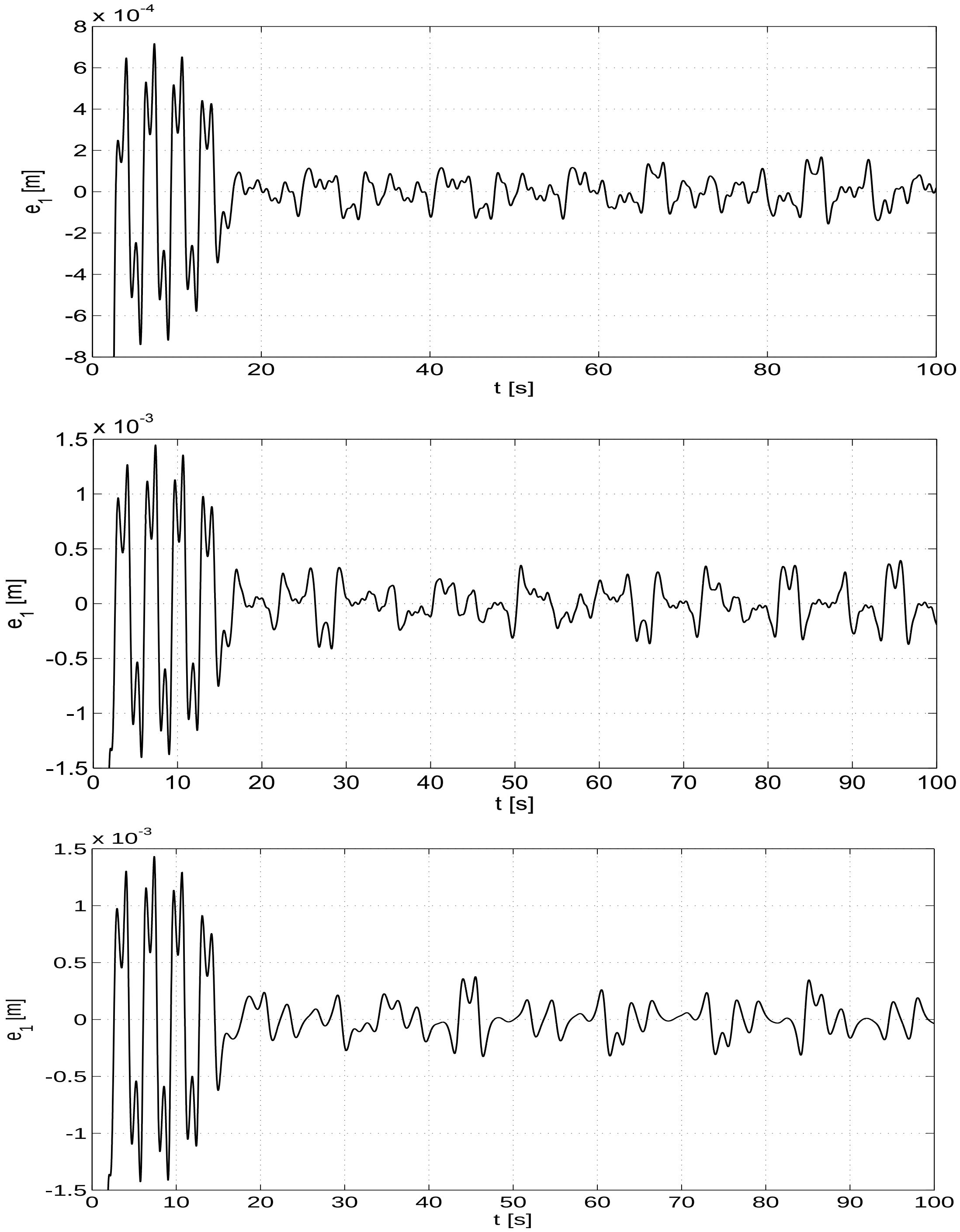


Figure 6.6: The tracking errors of the first state variables of the slave systems with controllers C_4, C_5, and C_6 ($e_1 = x_1 - y_1$); upper: C_4, middle: C_5, lower: C_6. It can be well seen that they all reduce the tracking error by more than two orders of magnitude, but the introduced controller C_4 gives 50% better result because of the second PD controller.

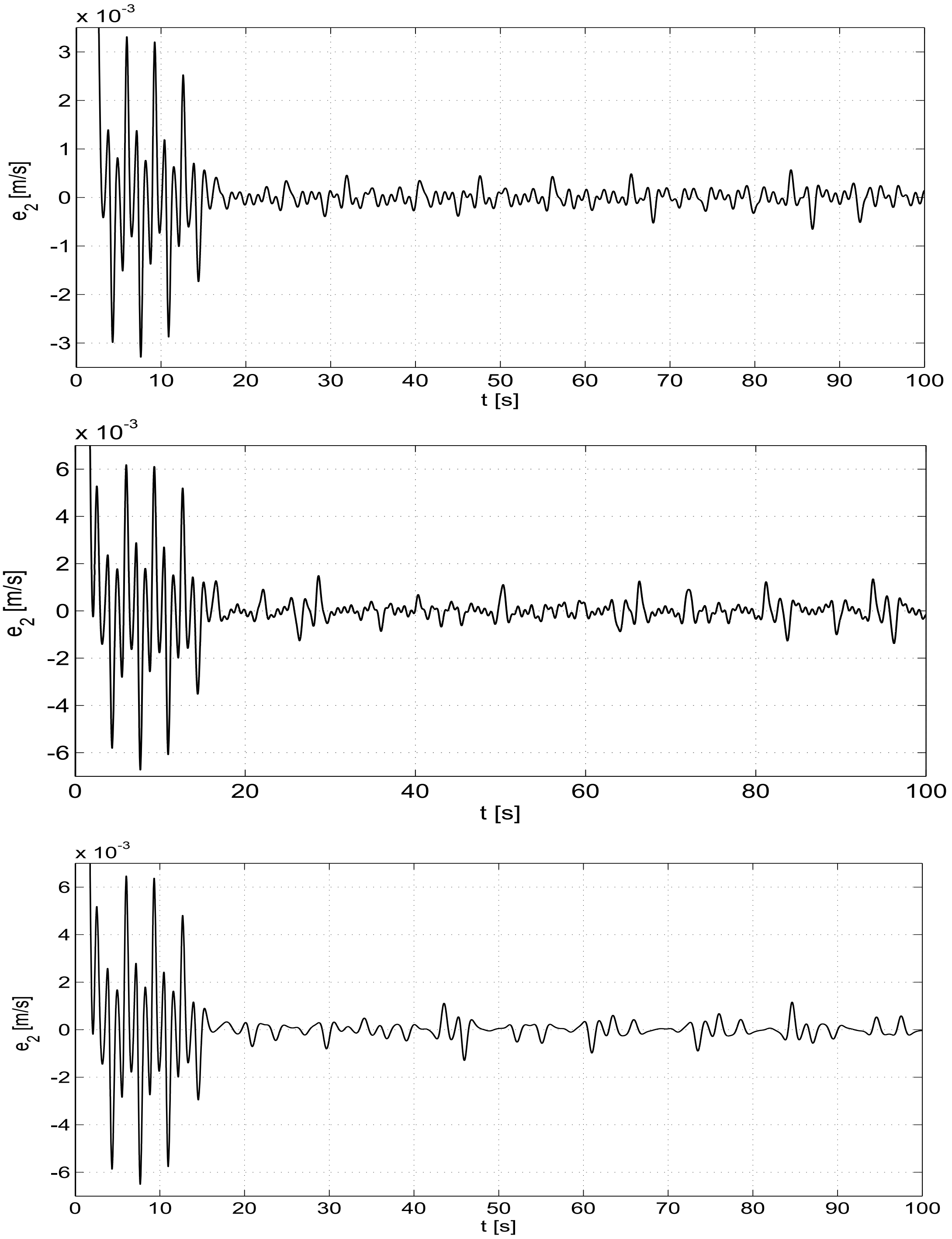


Figure 6.7: The tracking errors of the second state variables of the slave systems with controllers C_4, C_5, and C_6 ($e_2 = x_2 - y_2$); upper: C_4, middle: C_5, lower: C_6. It can be well seen that they all reduce the tracking error by more than two orders of magnitude, but the proposed controller C_4 gives 50% better result because of the second PD controller.

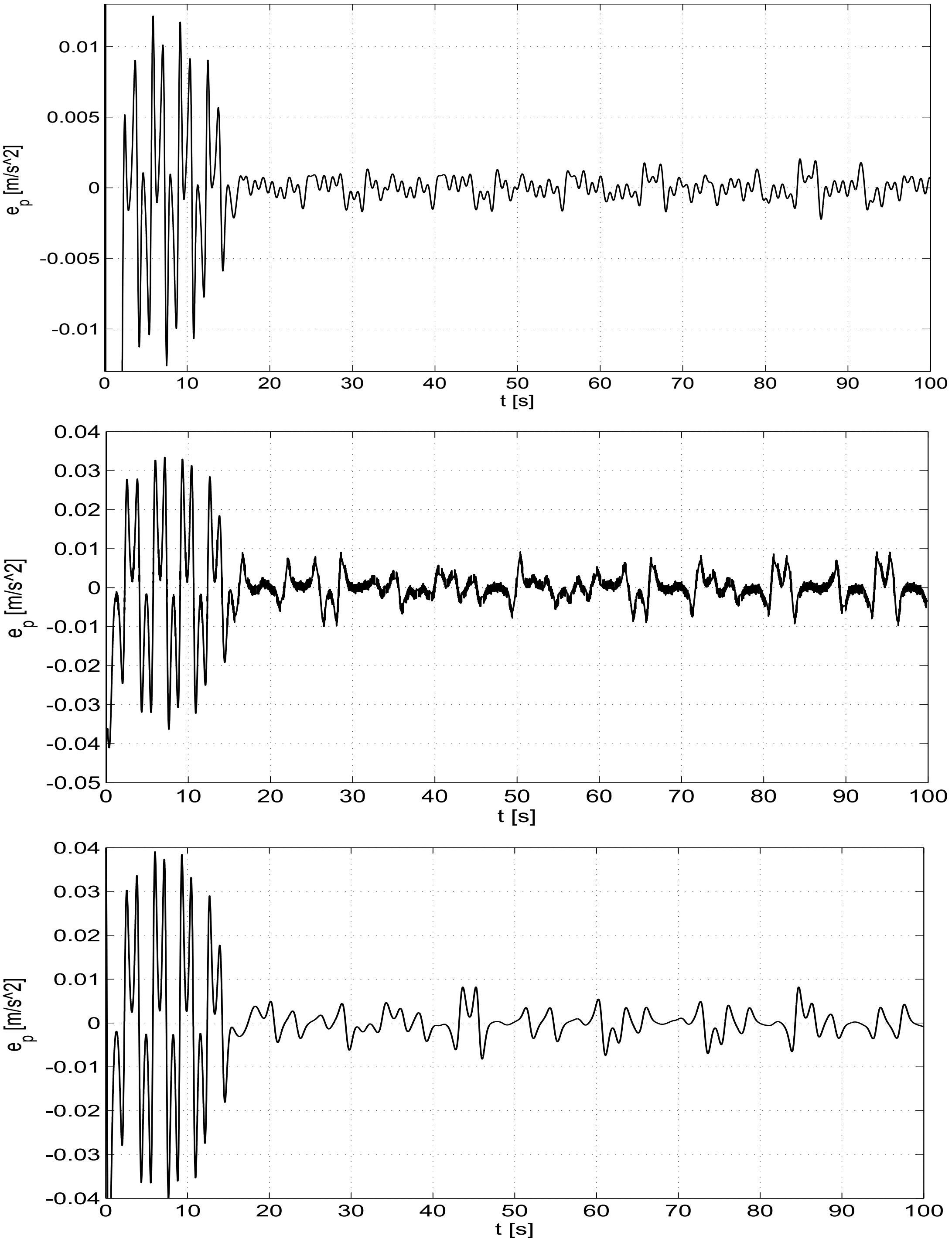


Figure 6.8: The difference between the desired and the realized response ($e_p = \dot{y}_2^d - \dot{y}_2^r$) with controllers C_4, C_5, and C_6 ($e_2 = x_2 - y_2$); upper: C_4, middle: C_5, lower: C_6. As it can be seen C_5 and C_6 generate similar results. The error achieved by the proposed "recalculated" controller C_4 is one third of the others because of the integrated second controller.

achieved by the previous versions of RFPT (by approximately 50%). However the second controller gives the burden of increased computational time, in case of using simple controllers, like the PID-type controllers, the drawback is negligible.

The result considered to be new has been published in journal paper [J2].

7

Fuzzy-type parameter tuning for Robust Fixed Point Transformations

In the previous chapter, a proposition is shown how to modify the structure of Robust Fixed Point Transformations. Since it does not prevent the stability problems, in this chapter, a new method is introduced how the results and the stability of the RFPT can be improved. The approach is based on the idea of tuning one of the three parameters of RFPT (see (4.6)). It is assumed that one of the three free parameters of RFPT does not have to be changed and there is a correspondence between the others. So if two parameters are fixed, only the third one has to be tuned. With this modification and a stabilization algorithm introduced in the next chapter, the stability of the RFPT-based controllers can be achieved.

7.1 The parameter tuning for RFPT

In this section, a possible parameter tuning strategy for RFPT is shown. According to the results of numerous simulations made for various physical systems, it becomes clear that two of the RFPT's parameters, B and K, can be easily determined and fixed. The problem is that fine tuning of A may be required for precise trajectory tracking. In [92] a relatively complicated tuning strategy is applied. However, the observation comes that if A remains inside of a range, then function G (where G represents G_1 or G_2 from

Chapter 4, or G_3 from Chapter 6) transforms its input closer to the right solution and the tuning of parameter A influences only the speed of the convergence.

Since parameter A always gets a small positive number as a value, the new tuning strategy is based on the idea of using a set of possible $\{A_1 = 10^{-\Delta}, A_2 = 10^{-2\Delta}, \ldots, A_m = 10^{-m\Delta}\}$ parameters instead of using a single one. Then, the possible A_i parameters are substituted into function G (the example is shown for G_2) and the function's possible outputs are calculated. The final output of RFPT H is got as a weighted sum of the possible outputs of function G:

$$H\left(r, r^d\right) = \sum_{i=1}^{m} w_i x_i \tag{7.1}$$

where

$$x_i := (r + K)\left[1 + B\tanh\left(A_i\left[f(r) - r^d\right]\right)\right] - K \tag{7.2}$$

The $w_i > 0$ weighting factors can be determined in many ways, e.g. they can be calculated through an optimization over $\mathbb{R}^m$. The idea of this thesis is applying a Gaussian-like function ϑ to calculate the weighting factors. ϑ can be explained as a fuzzy-like membership function of all of the values $i\Delta$ and its outputs $\tilde{w}_i$ as fuzzy-like membership values, where the used defuzzification method is the center of gravity [93]:

$$\tilde{w}_i := \vartheta(i, \zeta, \varsigma, \Delta) = \frac{1}{\varsigma^2 + (i\Delta - \zeta)^2} \tag{7.3}$$

and their normalized values

$$w_i := \frac{\tilde{w}_i}{\sum\limits_{j=1}^{m} \tilde{w}_j} \tag{7.4}$$

The block diagram of the proposed tuning method is shown in Fig. 7.1. The main idea lies in the modification of the remaining parameters of function ϑ after fixing its size (ς), and the number of- and step size between the possible A_i parameters (m and Δ, respectively). It can be done according to the following strategy. The center of the membership function (ζ) is set to an initial value and it is changed in every step. ϑ moves in time along the horizontal axis x with constant velocity $\dot{\zeta} > 0$. As the function moves in positive direction, the weights of the possible outputs (x_i) where $i\Delta < \zeta$

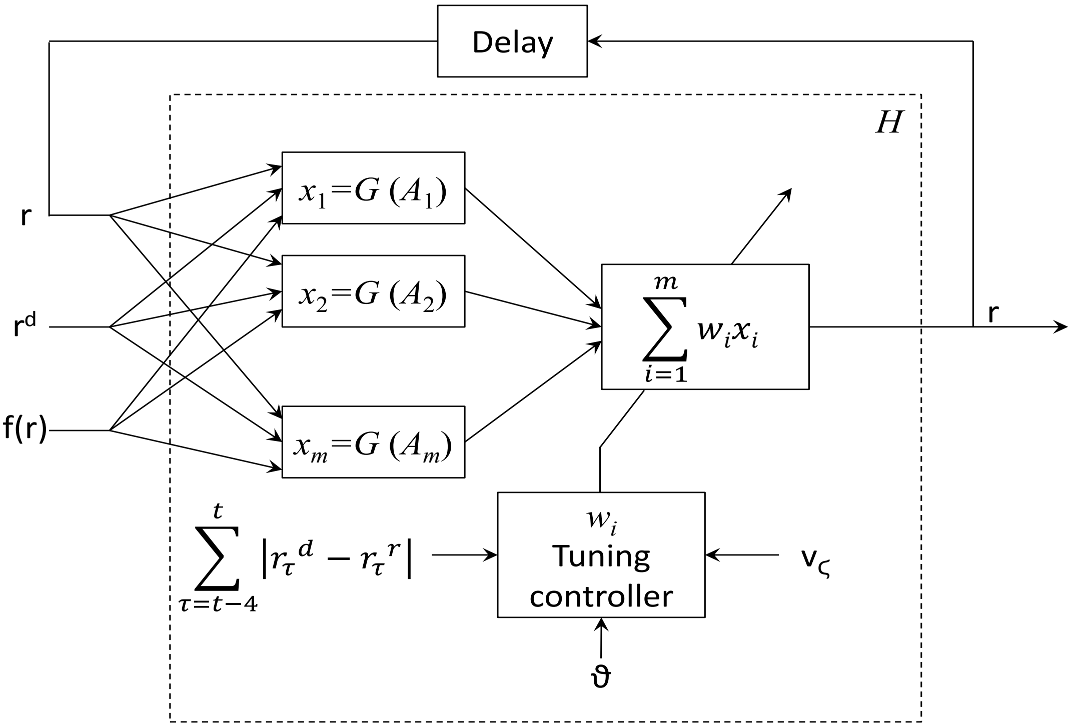


Figure 7.1: The block diagram of the proposed tuning strategy.

decrease while those with $i\Delta > \zeta$ increase. The greatest weight is given to that x_j for which $|j\Delta - \zeta|$ is minimal. The velocity of function ϑ can be determined as $\dot{\zeta} = \pm v_\zeta$ where $v_\zeta > 0$ is constant. If a given $\dot{\zeta}$ causes that the sum of the absolute response error ($|r^r - r^d|$, see Chapter 4) decrease within five control steps, it is kept. Otherwise, its sign is changed. Function ϑ is illustrated in Fig. 7.2.

7.2 Simulation Results

In the previous chapters, the RFPT method is used in that case if only one signal has to be transformed. If function G has multiple inputs, it has to be modified: e.g. instead of the absolute value, some kind of norm has to be used. To save computational time a different form of the function is used from this point: $\vec{h} := \mathbf{f}(\vec{r}) - \vec{r^d}$, $\vec{e} := \vec{h}/||\vec{h}||$, $\tilde{B} = B\tanh(A||\vec{h}||)$, so that $\mathbf{G}(r, r^d) = (1 + \tilde{B})\vec{r} + \tilde{B}K\vec{e}$. If $||\vec{h}||$ is very small, the approximation $\mathbf{G}(\vec{r}) = \vec{r}$ can be applied (since then the system is already in the very close vicinity of the fixed point).

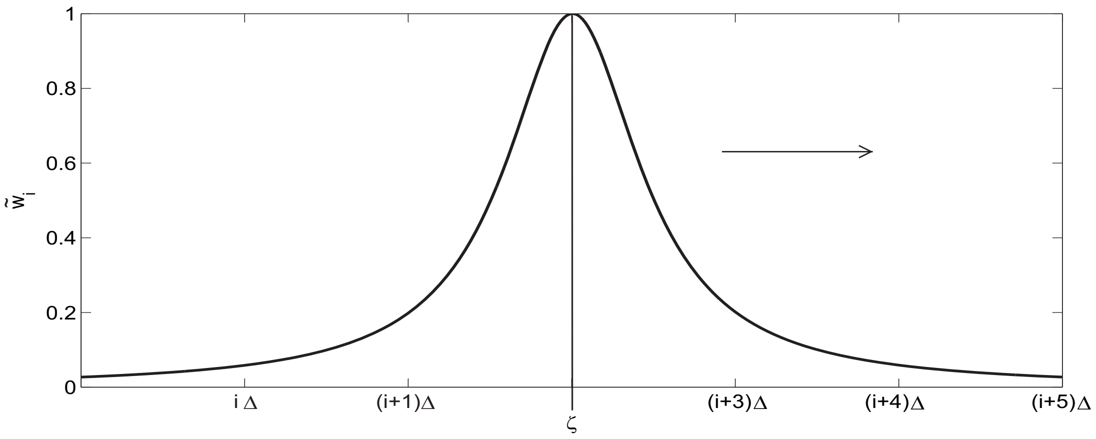


Figure 7.2: Function ϑ in case of $\varsigma = 1$.

In the following, some details are shown for the illustrative example. The above explained new parameter tuning is applied to control a cart plus double pendulum system (explained in Section 3.6). In the example a traditional PID controller is used: first without RFPT (controller C_1), then with basic RFPT (controller C_2), and finally, with RFPT and parameter tuning (controller C_3). The PID controller can be described as

$$\ddot{x_c}^{Des} = \ddot{x_c}^{N} + 3\Lambda\dot{e} + 3\Lambda^2 e + \Lambda^3 \int e \tag{7.5}$$

where $e = x_c^N - x_c$ denotes the tracking error, x_c^N corresponds to the nominal-, while x_c to the realized/simulated trajectory. The task is not to balance the pendulums, but to follow a third order spline function. The simulations are made by the use of SCILAB 5.1.1 and its graphical tool Scicos 4.2. In the present example, the following parameters are chosen without limiting the generality of the new result. The maximum step size of Scicos's integrator is set to $10^{-3}\,s$. The free parameters of RFPT are set to $K = -3.2 \times 10^4$, $B = 1$, $\{A_i := 10^{-8+i\varsigma} | i = 0, \ldots, 9\}$, $\varsigma = 0.4$, $\Delta = 2\varsigma$, and $|\dot{\zeta}| = 200$. The free parameter of the PID controller is $\Lambda = 5/s$. The system parameters are: $M = 20\,kg$, $m_1 = 8\,kg$, $m_2 = 8\,kg$, $L_1 = 2\,m$, $L_2 = 2\,m$, and $g = 9.81\,m/s^2$. In the control process an approximate model is used for the cart plus double pendulum system: it is assumed to have the same structure as the original system. The model parameters are, as follows: $\hat{M} = 10\,kg$, $\hat{m}_1 = 4\,kg$, $\hat{m}_2 = 6\,kg$, $\hat{L}_1 = L_1$, $\hat{L}_2 = L_2$, and $\hat{g} = g$. In the following, the simulation results are presented.

In Fig. 7.3 the three phase spaces achieved by controllers C_1, C_2, and C_3 can be compared. The upper figure reveals that with a simple PID controller (C_1) the trajectory tracking is very imprecise. If parameter A is fixed at 10^{-6} (C_2; middle), then the trajectory tracking is somewhere accurate but somewhere very poor. With parameter tuning (C_3; lower) the trajectory tracking is well achieved.

Some other results gained by controller C_3 are shown in the remaining figures: e.g. in Fig. 7.4 the tracking error can be followed. According to the author's observation, as function ϑ slithers along axis x, the weights w_i fluctuate. The fluctuation has a positive effect on the trajectory tracking as the weights adapt to the required changes. The process is illustrated in Figs. 7.5–7.7 for the presented simulation. When function ϑ moves in positive direction, the weights of the higher indexed values become bigger than that of the lowest ones (i.e. one of the lines is swapped with another one). Approximately stagnating periods can also be observed, as well as sessions when the function moves to negative direction.

7.3 Summary

In this chapter, a new parameter tuning strategy is proposed for Robust Fixed Point Transformations. The method is based on the idea of modifying one of function G's (G denotes G_1, G_2, or G_3) free parameters during simulations. The results show that though RFPT with fixed parameter can ameliorate the controllers' results in many cases, stable improvement is not always achievable. The proposed tuning technique can help this deficiency and precise trajectory tracking can be achieved when the parameter tuning is switched on.

The result considered to be new has been published in conference papers [C12, C13, L1, C18].

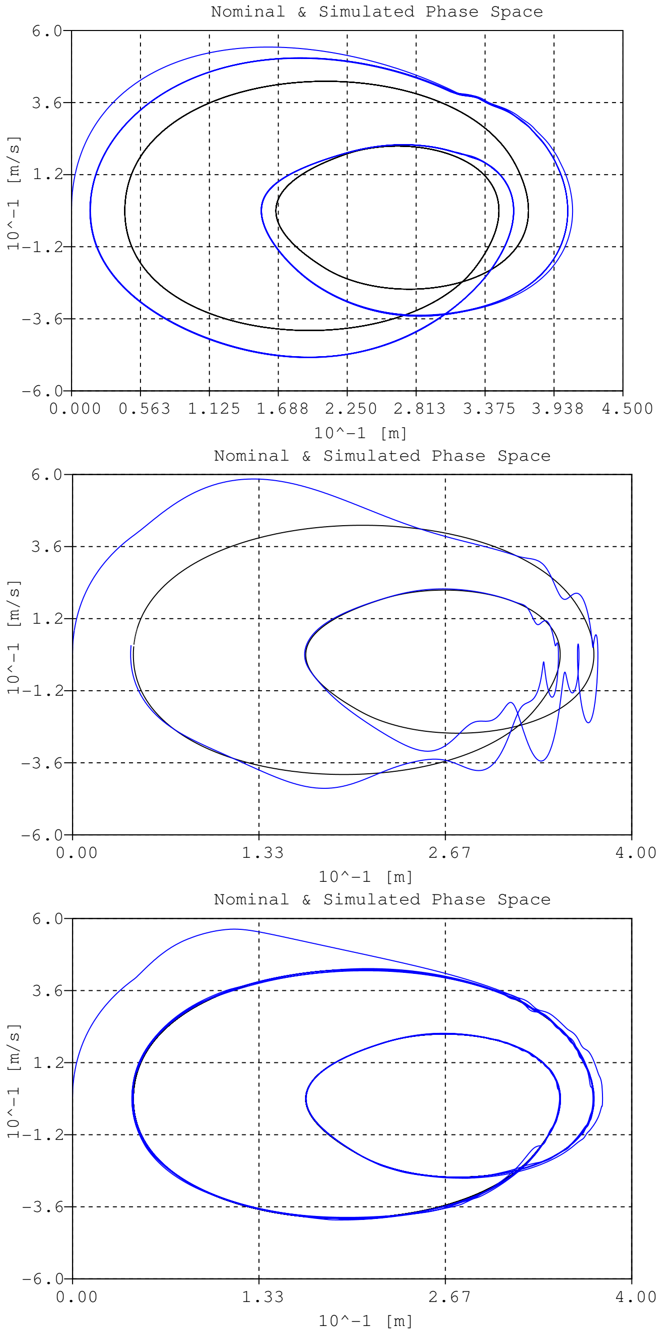


Figure 7.3: The nominal x_c^N vs. $\dot{x}_c^N$ (black line) and the simulated x_c vs. $\dot{x}_c$ (blue line) phase trajectories with controllers C_1 (upper), C_2 (middle), and C_3 (lower). RFPT improves the PID controller's results, but stable trajectory tracking is achieved only if the parameter tuning is switched on (C_3).

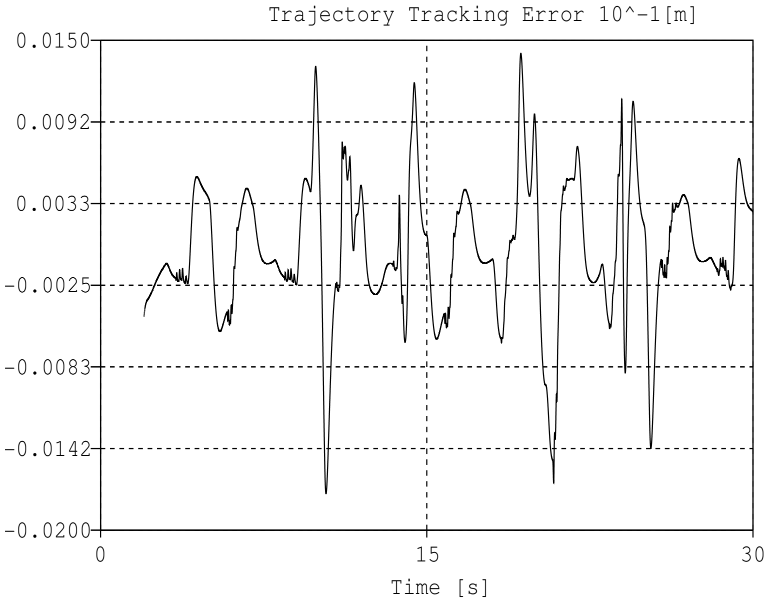


Figure 7.4: The tracking error achieved by controller C_3. The proposed parameter tuning causes tracking error reduction.

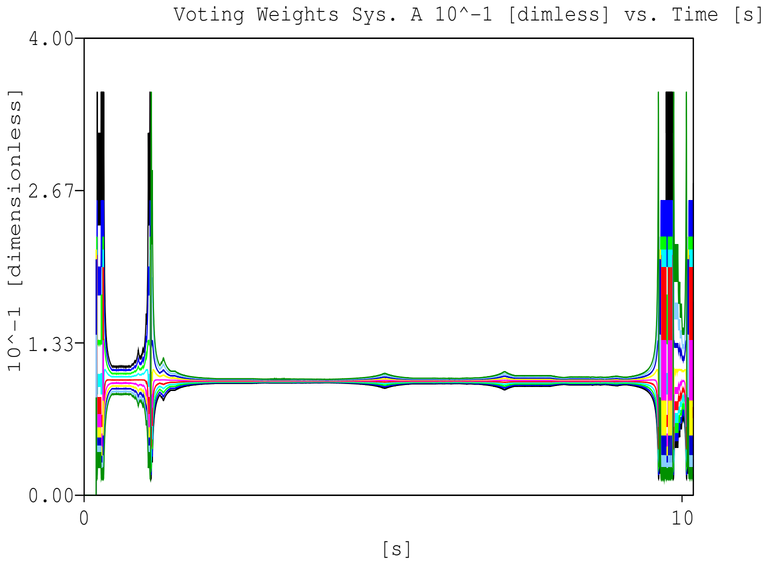


Figure 7.5: Excerpt 1: The fluctuation of the weights w_i in the first 10 seconds; 0: black, 1: blue, 2: green, 3: cyan, 4: red, 5: magenta, 6: yellow, 7: dark blue, 8: light blue, and 9: dark green. The realized trajectory nears to the nominal one, thus, the weights do not need to fluctuate.

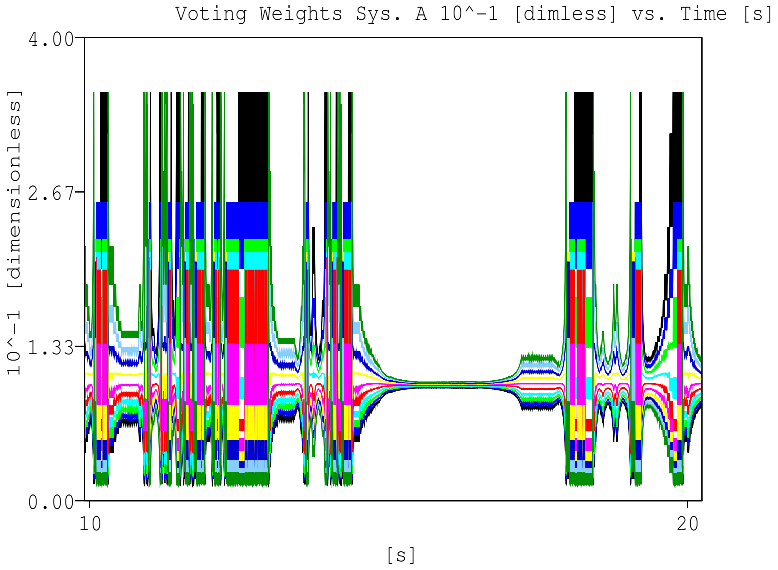


Figure 7.6: Excerpt 2: The fluctuation of the weights w_i in the second 10 seconds; 0: black, 1: blue, 2: green, 3: cyan, 4: red, 5: magenta, 6: yellow, 7: dark blue, 8: light blue, and 9: dark green. The weights significantly fluctuate causing more stable trajectory tracking.

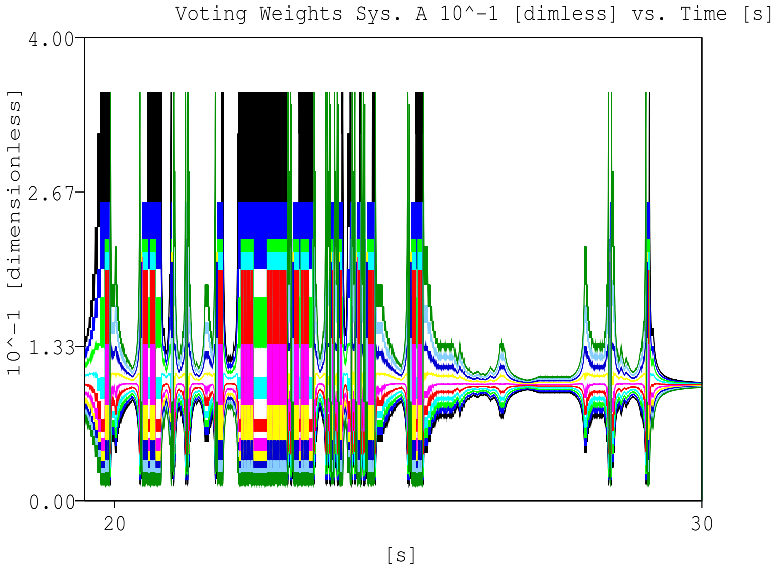


Figure 7.7: Excerpt 3: The fluctuation of the weights w_i in the last 10 seconds; 0: black, 1: blue, 2: green, 3: cyan, 4: red, 5: magenta, 6: yellow, 7: dark blue, 8: light blue, and 9: dark green. The weights significantly fluctuate causing more stable trajectory tracking.

8

VS-type stabilization for Robust Fixed Point Transformations

In Chapter 7, a parameter tuning algorithm is shown that improves the results and the stability of the Robust Fixed Point Transformations-based controllers. The parameter tuning provides the first step towards the stability of RFPT because besides all of its advantages it has a big drawback: It provides stability only of the Gaussian like shifting function ϑ remains in a certain interval around values $\log_{10} A_i$. If ϑ gets out of the interval then the RFPT-based controller becomes unstable and as a result the system to be controlled starts to chatter. The chattering can be dangerous and may result in the damage of the system.

In this chapter, a further development for RFPT is proposed which is able to guarantee the stability of the RFPT-based controller. With the combination of the previously suggested parameter tuning algorithm and the proposed VS-type stabilization method the controller will work in the following way: Although, it may happen that the shifting function ϑ gets far from values $\log_{10} A_i$ causing chattering, and temporal loose of stability, but afterwards the new method forces parameters A_i to move closer to ϑ and thus, the controller gains back its stability.

After this chapter, the applicability of RFPT in the soft-computing area is considered.

8.1 The stabilization algorithm

In Chapter 4, it is explained that the RFPT is based on a function (4.6) with the help of which a locally convergent sequence can be built. The sequence, depending on the free parameters of function G (where G represents G_1 or G_2 from Chapter 4, or G_3 from Chapter 6) can converge to two fixed points: a desired and a false one $(-K)$. When the free parameters of function G are not set correctly, the series converges to the false fixed point and the RFPT-based controller becomes unstable near to its desired solutions. In this case, when function G (the example is shown for G_1) gets an input very close to its desired fixed point, it transforms the input far away. So the next input will be far from the desired fixed point and will be out of the unstable zone, where starts to converge again to the desired fixed point. This behavior makes the controller similar to the Sliding Mode Controllers [94] for which the chattering effect is typical. The fluctuations observed in the control signal are in the order of magnitude of K, which is one of the three free parameters of (4.6) (it means huge chattering).

To avoid the stress caused by the fluctuation of the control signal, the following algorithm is proposed:

- Apply a sigmoid function σ_{VS} on the output of function H with properties $\sigma_{VS}(0) = 0$ and $\left.\frac{d\sigma(x)}{dx}\right|_{x=0} = 1$. Introduce a new parameter K_{vssm} so that $K \gg K_{vssm} > 0$ and use as $u_{t+1} = K_{vssm}\sigma_{VS}\left(\frac{H(u_t|u^d_{t+1})}{K_{vssm}}\right)$;
- Observe the change in the sign of the control signal with the help of a buffer u^{Buf}: If $u^{Buf}_{t+1} = \beta u^{Buf}_t + u_{t-2}u_{t-1}$ (where $\beta \in (0,1)$ is a free parameter) becomes negative, then chattering occurs, since the sign of the control force (or torque) varies in each control step;
- If $u^{Buf} > 0$ then apply the fuzzy-like parameter tuning explained in Chapter 7; If $u^{Buf} < 0$ then stop the parameter tuning and rigidly push the A_i parameters in the negative direction (however keeping $\forall i\, A_i > 0$), e.g. $\forall i\, A_i := 10^{-\epsilon}A_i$, where $\epsilon \in \mathbb{R}$ and $\epsilon > 0$, because function ϑ tries to refer to an output which cannot be calculated from the current values of A_i: .

As a result, the chattering is kept at bay. When the parameters $\{A_i\}$ decrease to the necessary extent, the convergence is restored and the chattering can completely be ceased.

Although, the above algorithm is basically shown for RFPT type 1 (see Section 4.3), it can be proven that the method can successfully be applied to all the other members of the RFPT family. The reason for this is that the causes of the chattering (and also the consequences) are similar in all of the cases.

8.2 Simulation results

In the following, some illustrative simulation results are shown for controlling a Φ^6-type Van der Pol oscillator (described in Section 3.4). The simulations are made in Scilab environment. In the example, the parameters are set as follows. The integrator maximum step size of SCICOS equals to $10^{-3}\,s$. For determining the desired state of the system the following PID controller is used:

$$\ddot{x}^d = \ddot{x}^{Nom} + 3\Lambda\dot{e} + 3\Lambda^2 e + \Lambda^3 \int e \tag{8.1}$$

where the tracking error is determined by $e = x^{Nom} - x^r$. x^{Nom} denotes the nominal trajectory, x^r stands for the realized trajectory, and Λ is a free parameter ($\Lambda = 12/s$).

In the example the parameters of RFPT are set to $K = 7000$, $K_{vssm} = 700$, $B = -1$, $\{A_i = 10^{-3+i\Delta}\, i = 0, ..., 2\}$, $\Delta = 0.05$, $\varsigma = 1$, and $\zeta = 1$. The parameter values of the oscillator are chosen as: $\mu_{vdp} = 0.4$; $\omega_0^2 = 0.46$; $\alpha = 1$; $\lambda_{vdp} = 0.1$; and $m_{vdp} = 1$.

The approximate model is designed to be very rough, thus, it can be described by the following equation:

$$Q = \hat{m}_{vdp}\ddot{x} + \hat{k}x \tag{8.2}$$

where Q denotes the control force applied on the system. The model parameters can be set freely, in the examples $\hat{m}_{vdp} = 2$, and $\hat{k} = 3\omega_0^2$ values are chosen. The nominal trajectory is determined by a quasi-sinusoidal function as

$$x^{Nom}(t) = C\sin(\omega t)\tanh(\omega t) + 1 \tag{8.3}$$

with $C = 1.5$ and $\omega = 3$.

During the simulations, two controllers are compared: one without RFPT (C_1), and one with RFPT and parameter tuning extended with chattering reduction (C_2). When

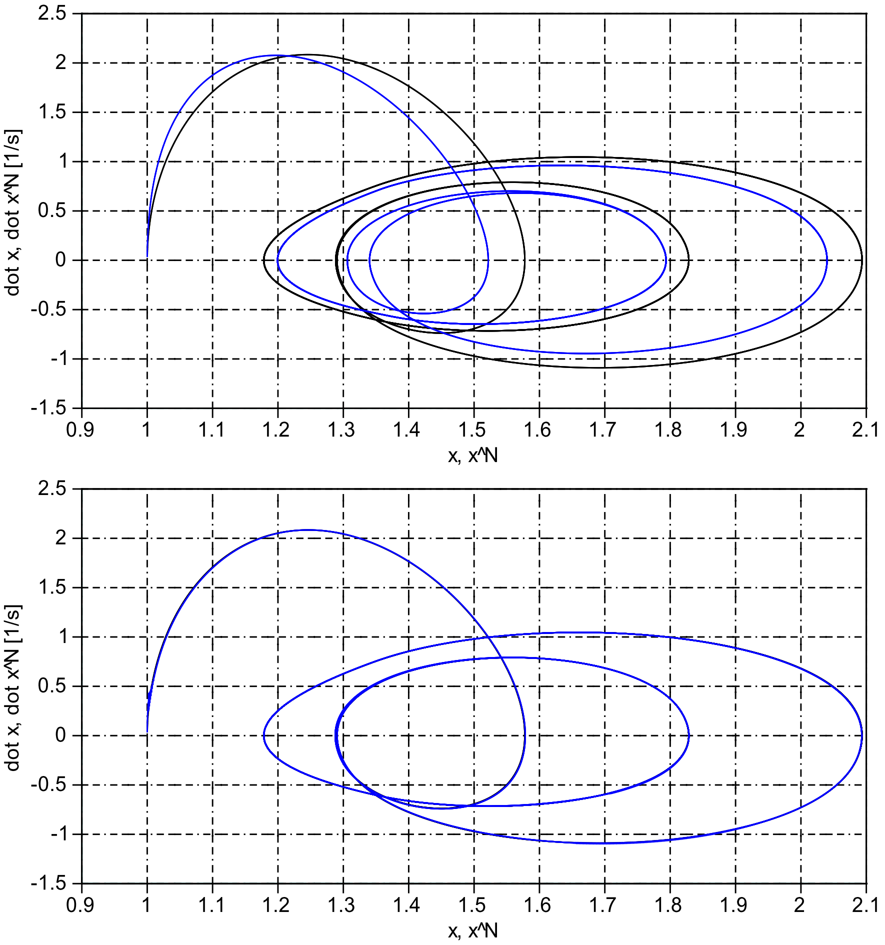


Figure 8.1: The nominal (black) and the realized (blue) phase space achieved by controllers C_1 (upper) and C_2 (lower). The trajectory tracking is much better when using C_2.

RFPT is applied, chattering occurs possibly causing the damage of the system. With the application of the proposed algorithm it can be avoided.

In the first figure (see Fig. 8.1) the nominal and simulated phase space using controllers C_1 (upper) and C_2 (lower) can be compared. With C_1 only imprecise trajectory tracking can be achieved because the model approximation is too rough. With C_2 the trajectories run together.

Figure 8.2 illustrates the applied torques. As it can be seen, with controller C_2 chattering occurs three times however it is relaxed in very short time. The chatters can be seen in details in Fig. 8.3.

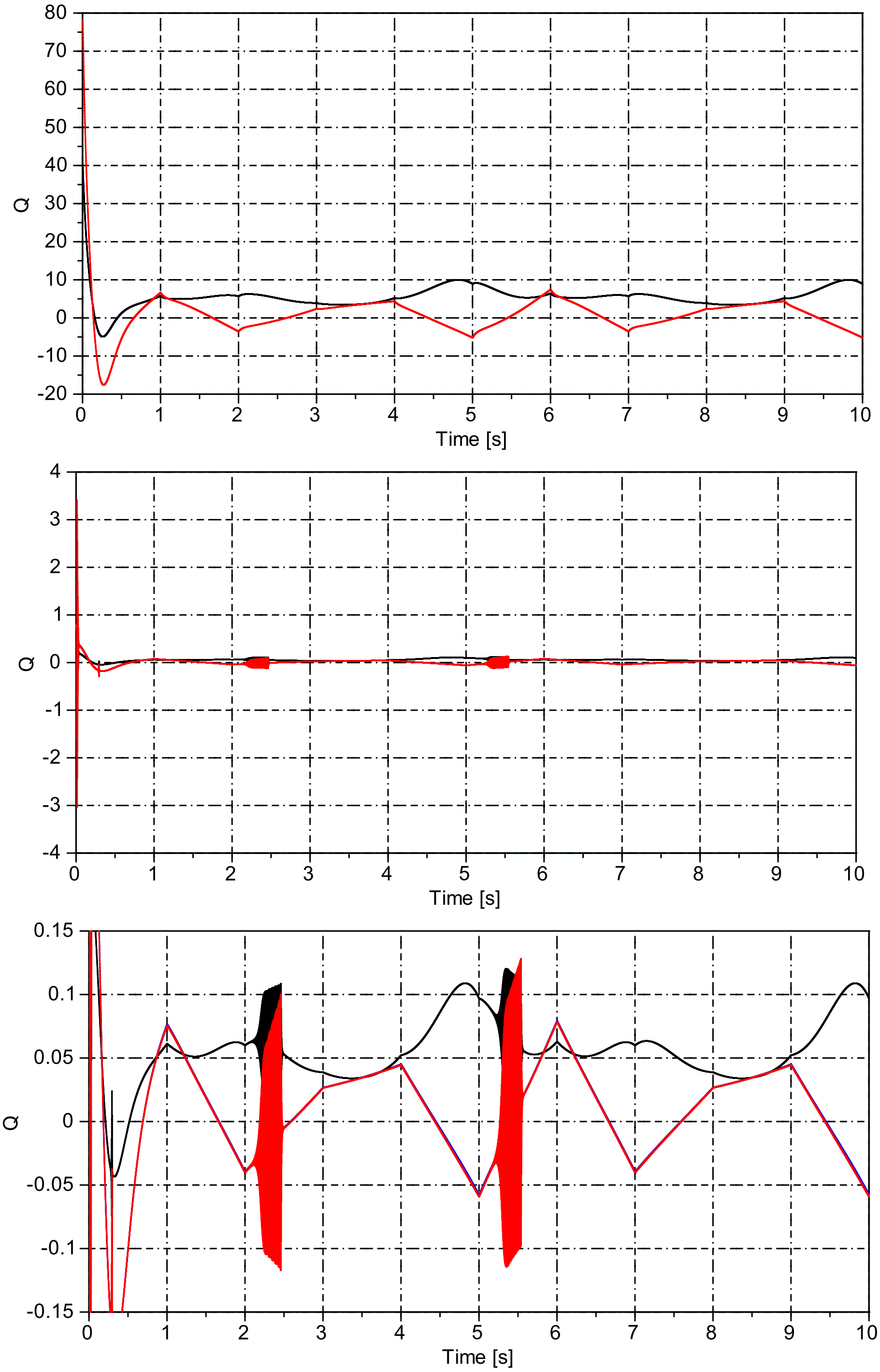


Figure 8.2: Approximate (Q^d_{appr}; blue), realized ($Q = G(Q^d_{appr})$; black), and recalculated ($h(Q)$; red) torques achieved by controllers C_1 (upper) and C_2 (normal - middle; zoomed - lower). In the lower figures chattering occurs three times and stopped in short time.

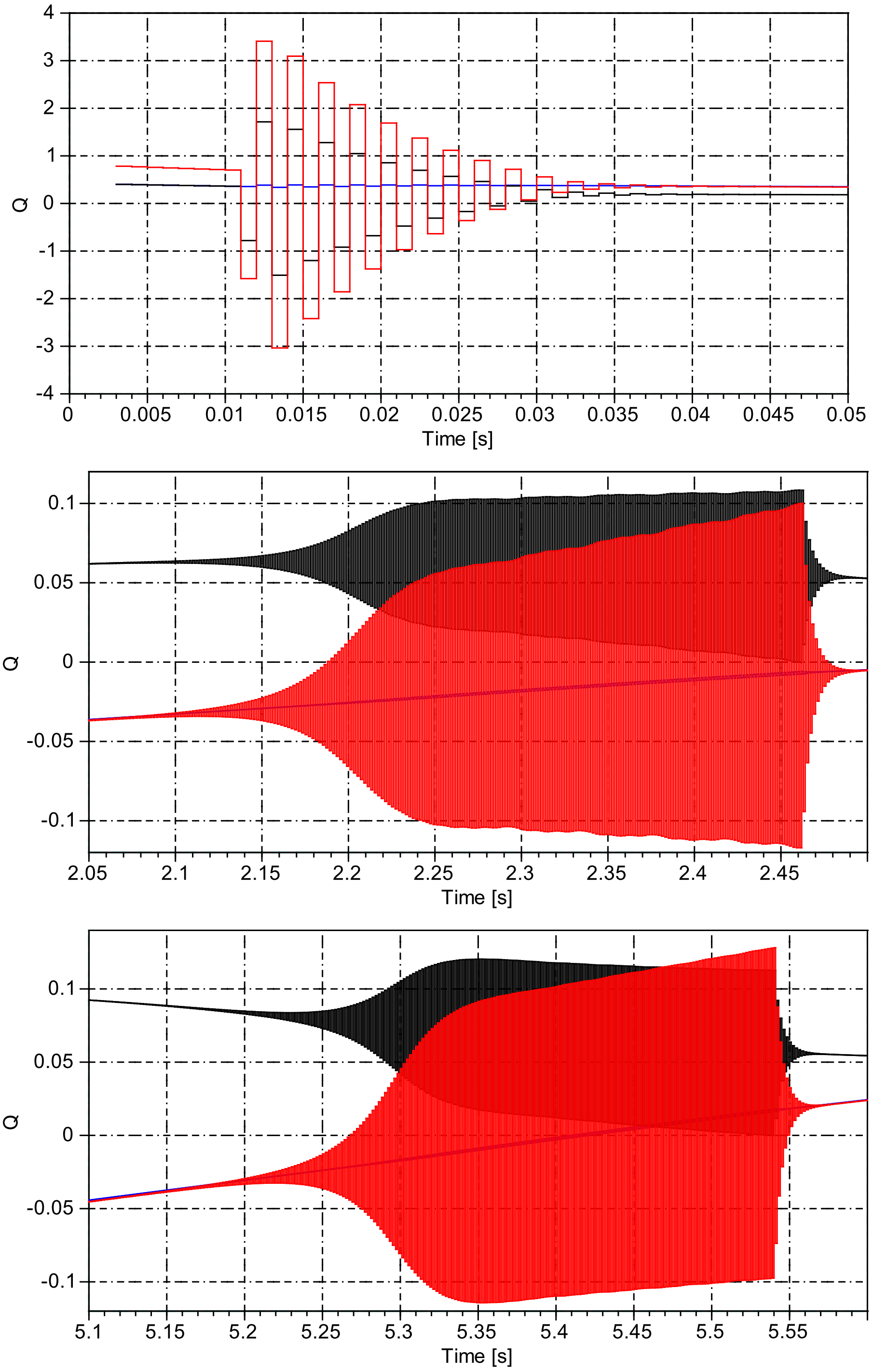


Figure 8.3: Approximate (Q^d_{appr}; blue), realized ($Q = G(Q^d_{appr})$; black), and recalculated ($h(Q)$; red) torques achieved by controller C_2, in details. Chattering occurs, but the proposed algorithm relaxes and stops it in short time.

The last figure (see Fig. 8.4) illustrates the tracking errors achieved by the two controllers. The initial balancing period is not shown. As it can be seen, though chattering affects the system, the tracking error achieved by C_2 is significantly smaller.

8.3 Summary

Robust Fixed Point Transformations make the controllers similar to the Sliding Mode Controllers. The reason of the similarity is that RFPT is based on a function which can locally converge to the ideal solution. Although, if the RFPT's free parameters are not set properly, the RFPT-based controllers can loose their convergence and chattering may occur. Although, in this chapter, it is shown that though RFPT can make the controllers unstable for very short periods, it can still improve a controller's results. A simple algorithm is introduced to minimize the fluctuation of the system and thus, preventing it from damages. As a result, stability of the RFPT-based controllers is achieved and though, function H is not always convergent, RFPT can reduce the tracking error achieved by the original controller significantly.

The result considered to be new has been published in conference paper [C19].

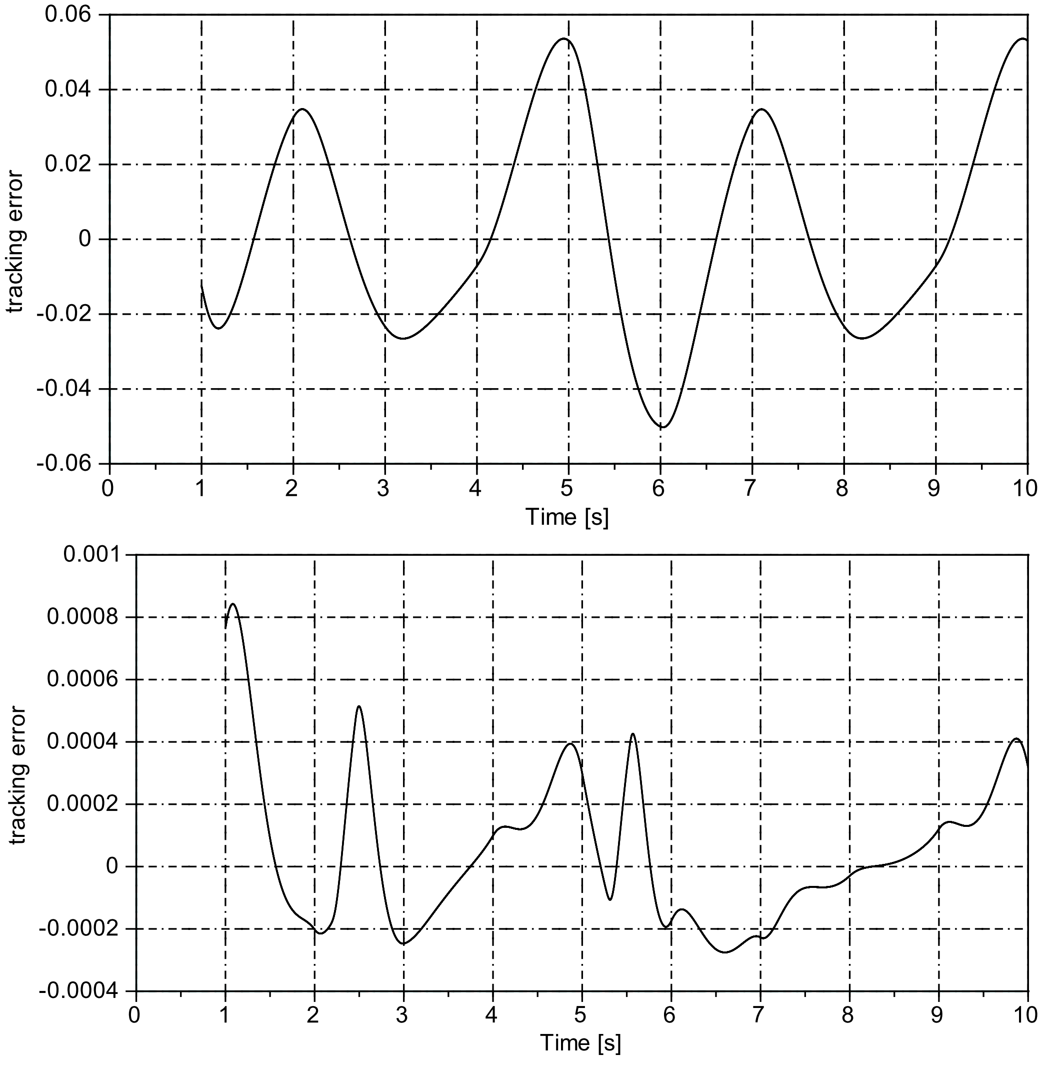


Figure 8.4: The tracking error achieved by controllers C_1 (upper) and C_2 (lower). With C_2 the tracking error is significantly smaller (even with the chattering effect).

9

Fuzzyfied Robust Fixed Point Transformations

In the previous chapters, two methods can be seen to make Robust Fixed Point Transformations-based controllers stable. In this one and the next chapter, a new aspect of RFPT is investigated: how it can be used to improve soft computing-based controllers: in the first case a fuzzy logic controller, then a neural network controller. The approaches are shown via the control of a cart-pendulum, where an approximate model is constructed for the system and an RFPT-based soft computing controller (in this chapter, a fuzzy logic controller) is designed to balance the pendulum based on the model approximation. In the next chapter, similar efforts are taken with a neural network controller.

9.1 Introduction

Fuzzy control methodologies have emerged in the recent years as promising tools to solve nonlinear control problems. The Fuzzy approach was first proposed by Lotfi A. Zadeh in 1965 when he presented his seminar paper on fuzzy sets [39]. Zadeh showed that fuzzy logic, unlike classical logic, can handle and interpret values between false (0) and true (1). One of the most successful application areas of fuzzy theory proved to be fuzzy logic control (FLC), because FLC systems can replace humans in performing certain tasks with high risk level, for example in the control of power plants [1].

The other reason for applying fuzzy techniques in control is their simple approach (describability with human language) which provides the handle of uncertainty and the use of heuristic knowledge for nonlinear control problems. In very difficult situations, where the plant parameters are disturbed or when the system is too complex to be described by exact mathematical models, adaptive schemes have to be used to gather data and adjust the control parameters automatically. Based on the universal approximation theorem [43] and by incorporating fuzzy logic systems into adaptive control schemes, a stable fuzzy adaptive controller is suggested in [44] which was the first controller being able to control unknown nonlinear systems. Later, many adaptive fuzzy control approaches have been developed for such systems (see e.g. [45, 46]).

In this chapter, to prove that fuzzy logic controller can be combined with Robust Fixed Point Transformations, a new RFPT-based FLC is introduced. In the example, the proposed new controller is applied to supervise the balancing of an inverted pendulum on the top of a cart (described in Section 3.5). FLC has successfully been used in the inverted pendulum-problem, e.g. in [62]. For comparison, in this thesis, the same controller is used. According to the results, the RFPT-based FLC outperforms the traditional fuzzy logic controllers and significantly reduces the necessary balancing time. The more extreme situation is chosen (e.g. if the initial angle of the pendulum is very high and no friction is assumed) the bigger difference is got between the balancing times of an RFPT-based and a traditional Fuzzy Logic Controller.

9.2 Extending Fuzzy Logic Control with RFPT

The mathematical basics of FLC are detailed in Subsection 2.3.1. The applied fuzzy logic controller is taken from [62]. The controller has two inputs: the angle and the angular velocity of the pendulum. From these values FLC calculates the desired torque (F) for the cart. Then, as a part of the control process, the desired $\ddot{x_c}^D$ is determined from the desired torque with the approximate model, so the controller has to deal with double model approximation. Figures 9.1-9.3 show the membership functions of the FLC's inputs and output, while in Figs. 9.4 and 9.5 the rule base and the rule surface can be seen.

Since so far the Robust Fixed Point Transformations method has been applied to improve only classical controllers, it is important to show, that it can cope with

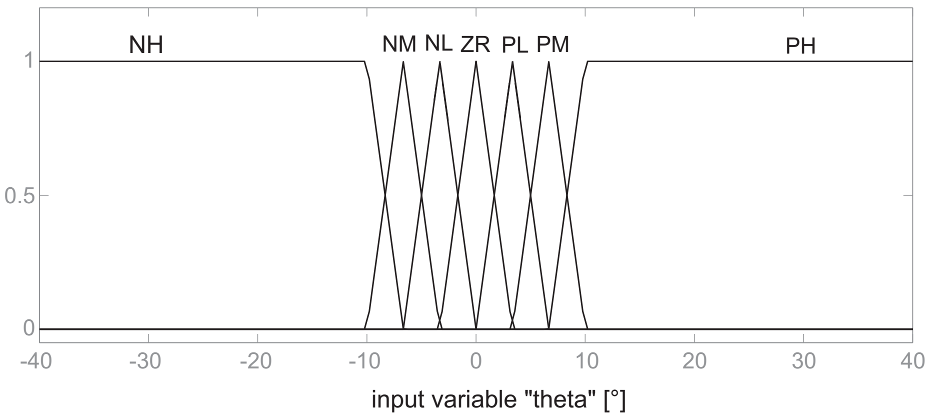


Figure 9.1: Membership functions for θ.

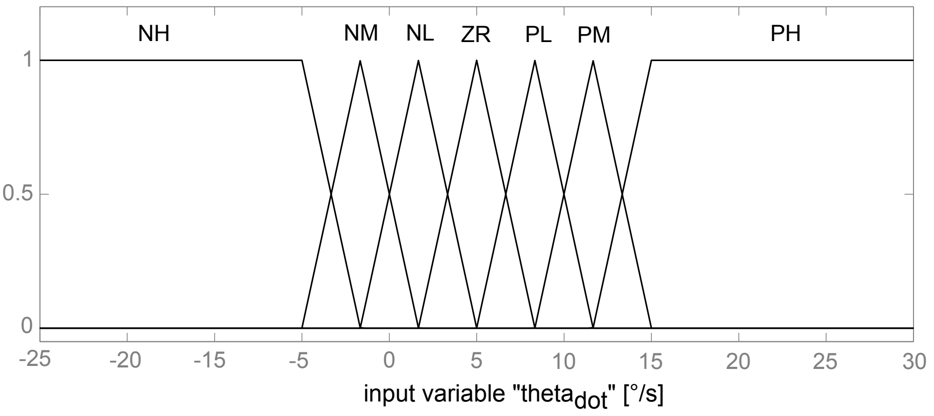


Figure 9.2: Membership functions for $\dot{\theta}$.

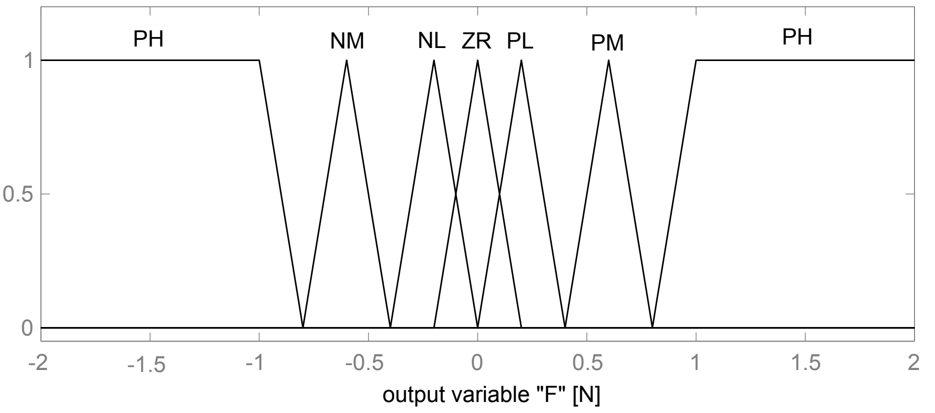
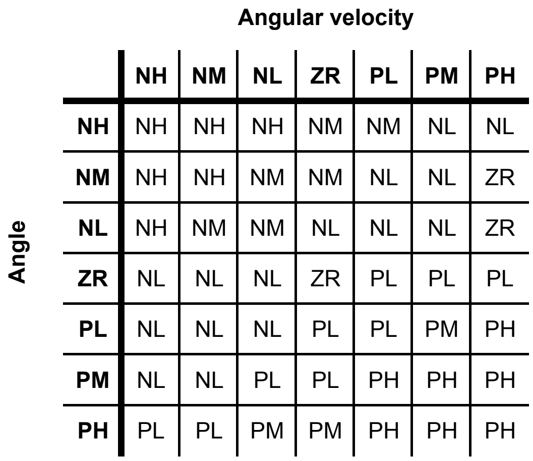


Figure 9.3: Membership functions for F.



Angular velocity

Angle

	NH	NM	NL	ZR	PL	PM	PH
NH	NH	NH	NH	NM	NM	NL	NL
NM	NH	NH	NM	NM	NL	NL	ZR
NL	NH	NM	NM	NL	NL	NL	ZR
ZR	NL	NL	NL	ZR	PL	PL	PL
PL	NL	NL	NL	PL	PL	PM	PH
PM	NL	NL	PL	PL	PH	PH	PH
PH	PL	PL	PM	PM	PH	PH	PH

Figure 9.4: Rule base for the Fuzzy Logic Controller.

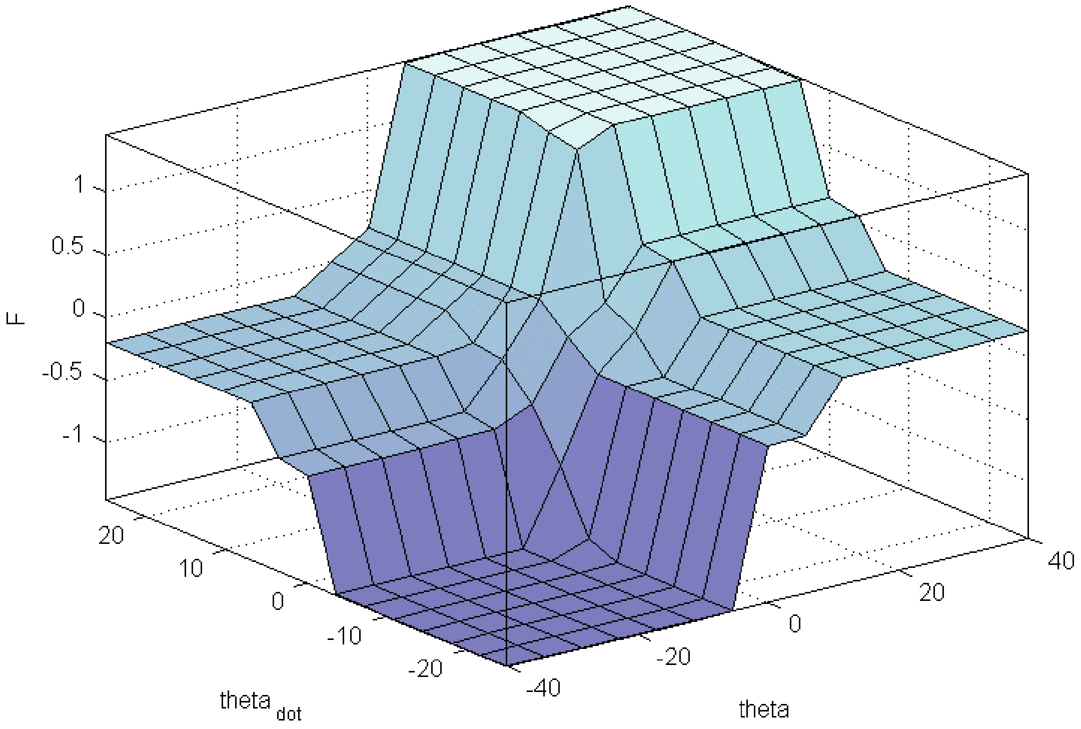


Figure 9.5: Control surface of the FLC.

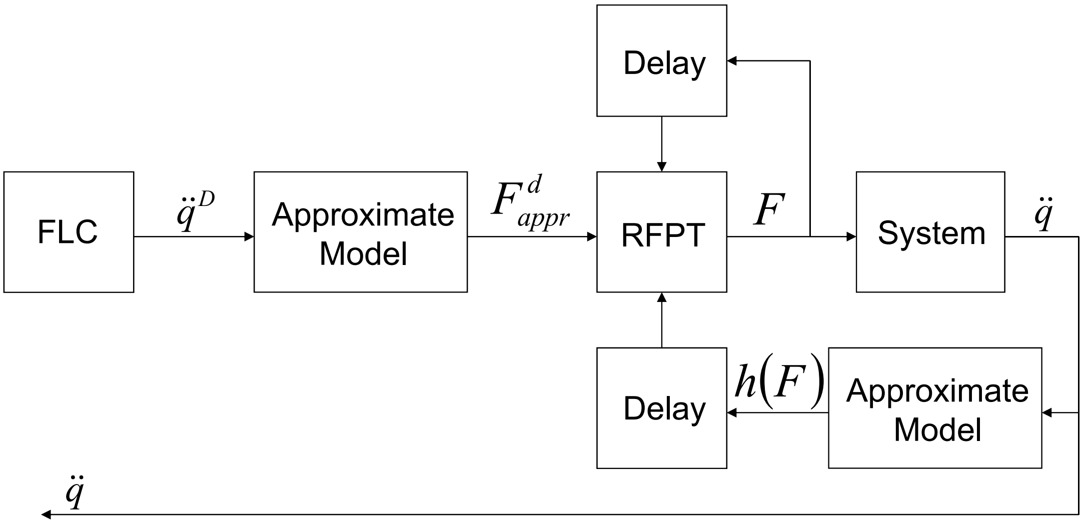


Figure 9.6: The block scheme of the RFPT-based fuzzy logic controller, where $\ddot{q} = \left[\ddot{x}_c, \ddot{\theta}\right]$.

recent controllers, too. In the following, it is shown that fuzzy logic controller, which is one of the popular soft computing-based controllers, can be improved with RFPT. Furthermore, the analogy can be applied to all the other types of SC based controllers. In the examples shown in this chapter, the FLC is extended with the first type RFPT (see Section 4.3). Figure 9.6 shows the block diagram of the proposed scheme.

9.3 Simulation results

In this section, some illustrative simulation results are presented. The simulation task is balancing a pendulum on the top of a cart (the details of the system are described in Section 3.5). The simulations are made by the Matlab-Simulink package. Without limiting the generality of the new result, in the examples, the following parameters are used: $M = 1.5kg$, $m = 0.5kg$, $L_p = 0.1m$, $g = 9.8m/s^2$, and $b_{cp} = 0.2kg/s$.

The approximate model is assumed to have the same structure as the original cart-pendulum system. The approximate model parameters are set to $\hat{M} = 1.2kg$, $\hat{m} = 0.8kg$, $\hat{L_p} = 0.09m$, $\hat{g} = 10m/s^2$ and $\hat{b}_{cp} = 0.1kg/s$. The initial values for the state variables are chosen as $x_c = 0$ and $\theta = 15°$.

During the simulations two controllers are compared: FLC without RFPT (C_1), and FLC with RFPT (C_2). In the first example, the simulation runs without measurement noise. In Figs. 9.7 and 9.8, the angle of the pendulum and the position of the cart,

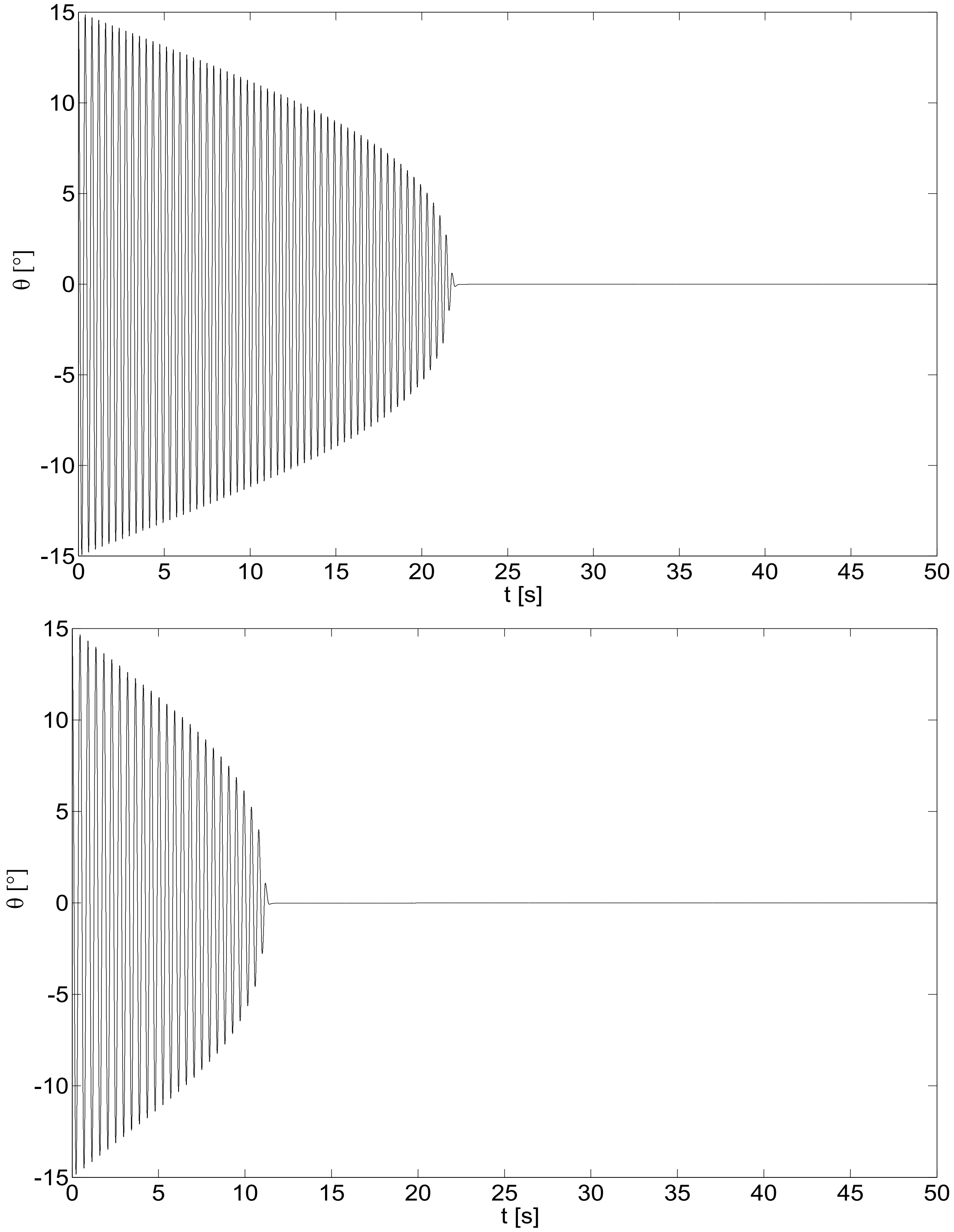


Figure 9.7: The angle of the pendulum with controllers C_1 (upper) and C_2 (lower), without noise. With C_2 the stabilization is achieved twice quicker.

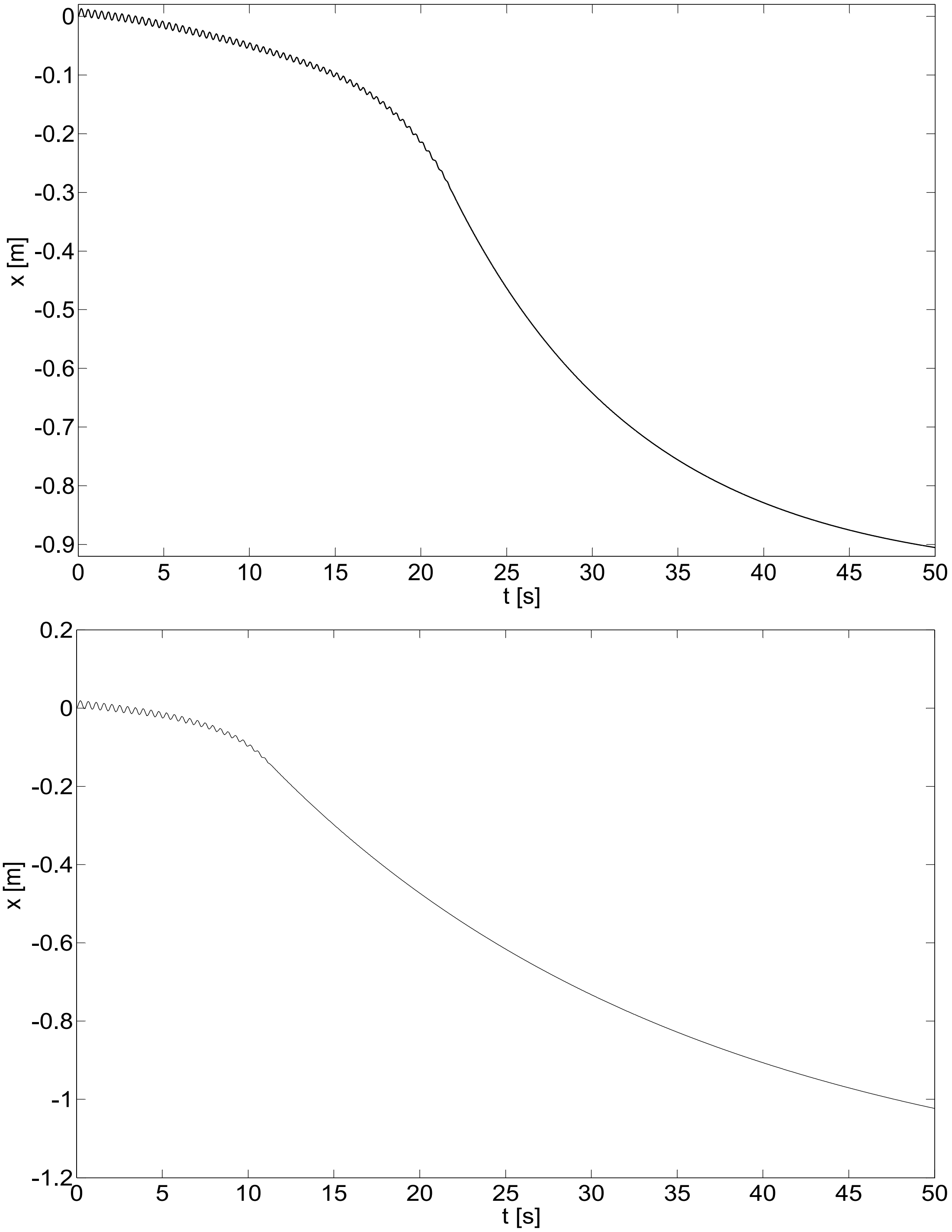


Figure 9.8: The position of the cart with controllers C_1 (upper) and C_2 (lower), without noise. The position stabilization is slow in both cases because of the double model approximation.

achieved by the two controllers, can be compared, respectively. The parameters for the transforming function built in the RFPT (G_1) are set to $B = -1$, $A = 9 \times 10^{-6}$, and $K = -10^5$. As the results show, controller C_2 can stabilize the pendulum in twice shorter time than C_1.

In the second example, some kind of measurement noise is assumed which is represented by two sinusoidal waves added to the real system ($\ddot{x}_c$ and $\ddot{\theta}$, respectively). These waves have the amplitude of $0.4m$ and $0.5m$ and frequency of $4Hz$ and $2.7Hz$, respectively. Figures 9.9 and 9.10 show that controller C_2 is twice better again and while it is able to slowly stabilize the position of the cart, controller C_1 cannot.

For the effective comparison, the results of a third simulation are included, too. The third simulation has very extreme initial conditions: no friction is assumed ($b = \hat{b} = 0$), and the initial state variables are $x_c = 0$ and $\theta = 72°$. Figure 9.11 illustrates the variation in the angles of the pendulum achieved by the two controllers. Controller C_1 gains only 2° improvement in 100 seconds. In contrast, controller C_2 can completely stabilize the pendulum in less than 60 seconds. Because of the model approximation and that no friction is assumed, the position of the cart, shown in Fig. 9.12, is not stabilized in neither case.

9.4 Summary

Soft computing-based controllers are widely used, very popular controller types. In this chapter, an extended FLC scheme is introduced, which offers an opportunity to ameliorate an existing and well set controller's results. As an example, the presented new Robust Fixed Point Transformation-based fuzzy logic controller is applied to balance a cart-pendulum system. The results show that in simple cases the RFPT can bisect the balancing time of the FLC. Furthermore, the more extreme situation is chosen the bigger difference can be got between the balancing times of an RFPT-based and a traditional fuzzy logic controller.

The result considered to be new by the author has been published in conference paper [C20].

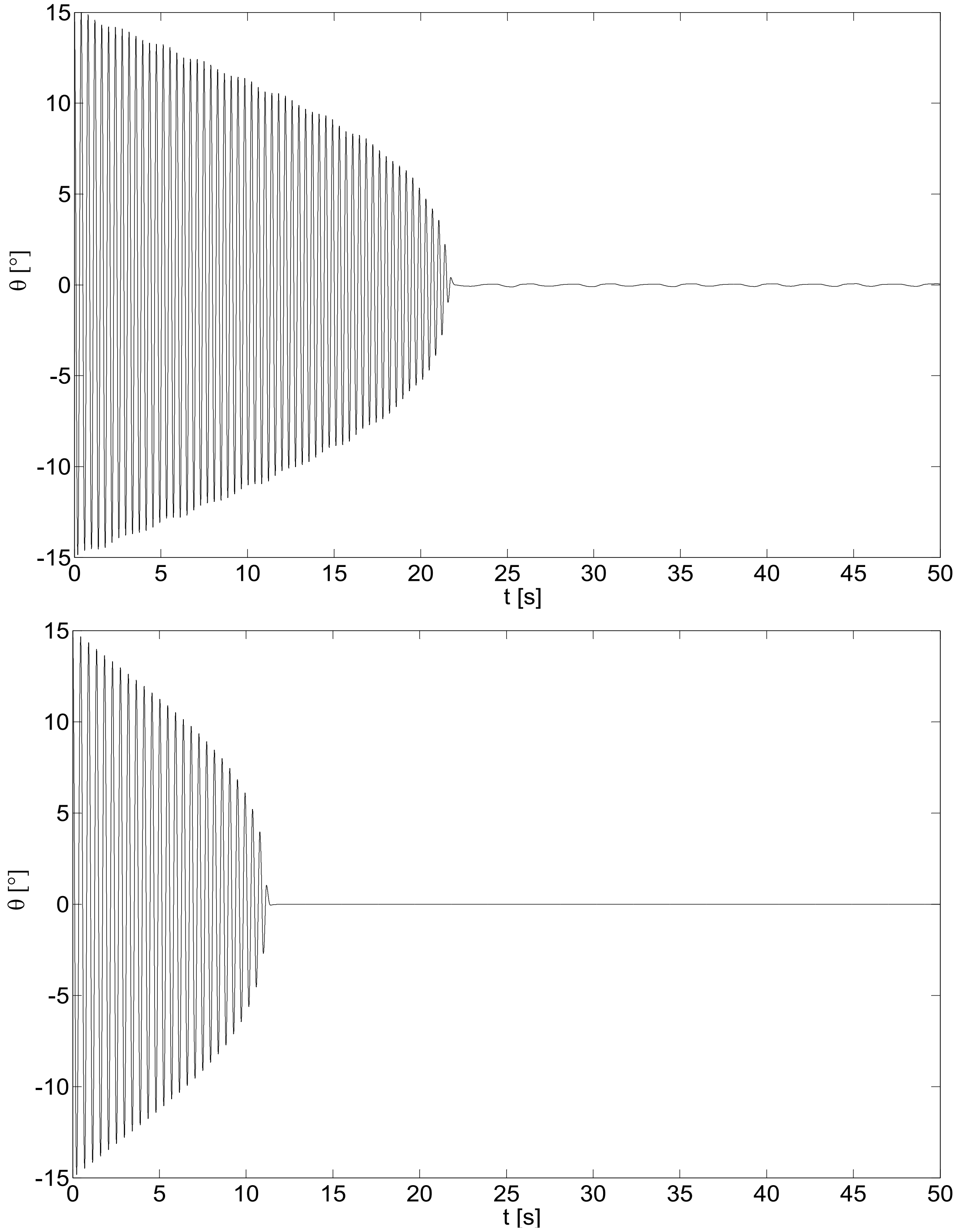


Figure 9.9: The angle of the pendulum achieved by controllers C_1 (upper) and C_2 (lower), noise added. Using C_2 the stabilization is achieved in twice shorter time.

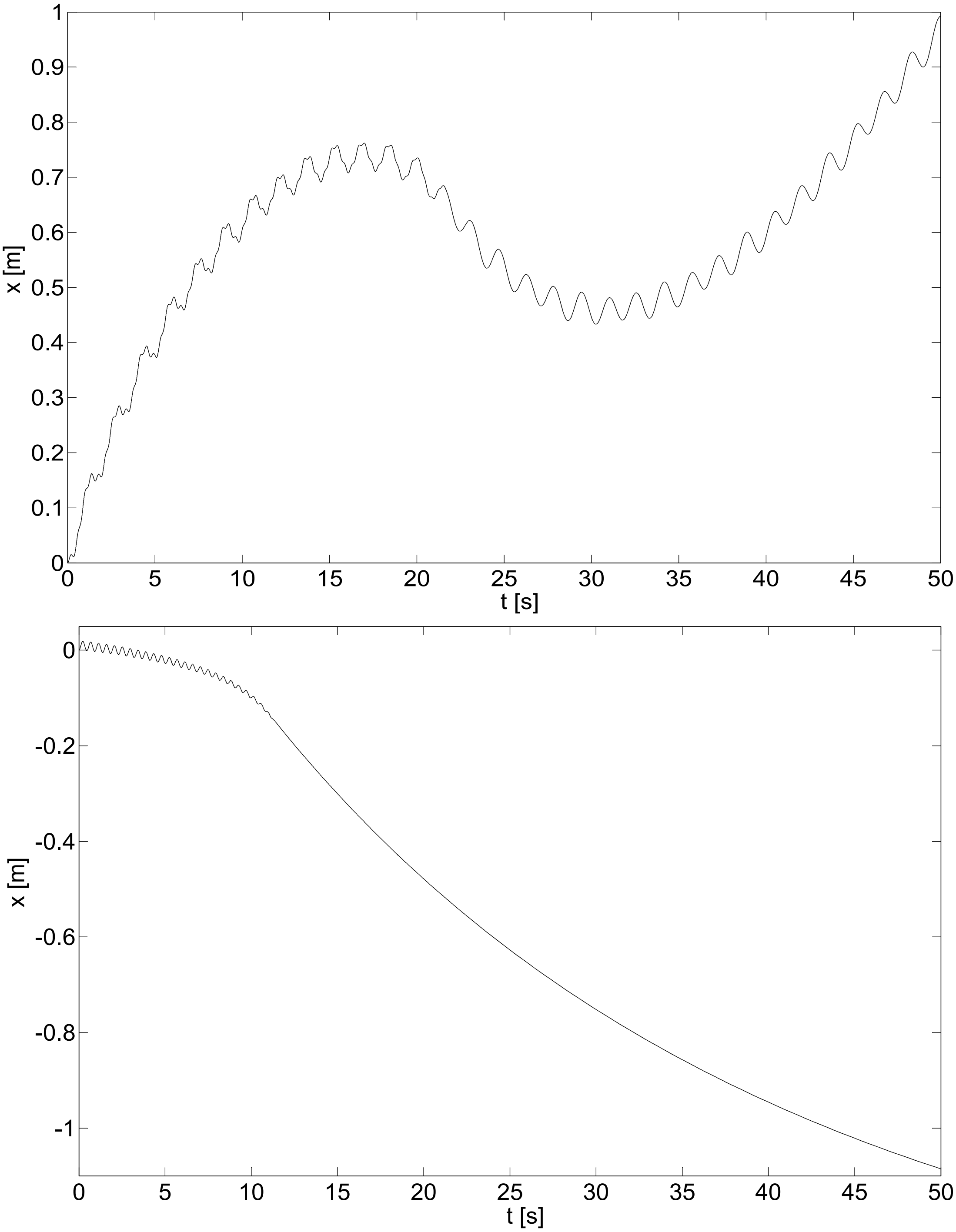


Figure 9.10: The position of the cart using controllers C_1 (upper) and C_2 (lower), with noise. The position stabilization is achieved only with C_2.

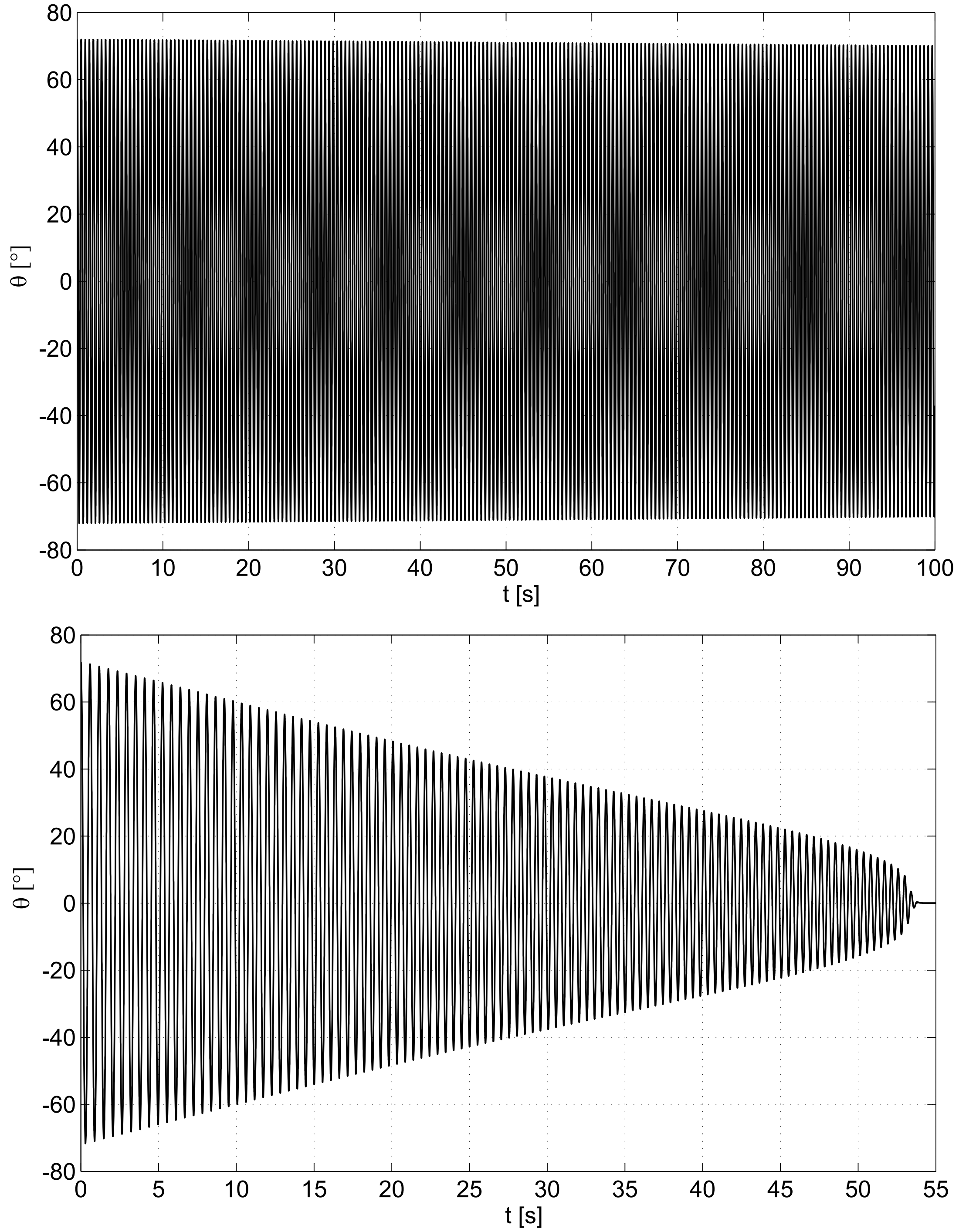


Figure 9.11: The angle of the pendulum using controllers C_1 (upper) and C_2 (lower), without noise, starting with very wide initial angle. The improvement of θ achieved by C_1 is only 2° in 100 seconds, while C_2 succeeds in less than 55 seconds.

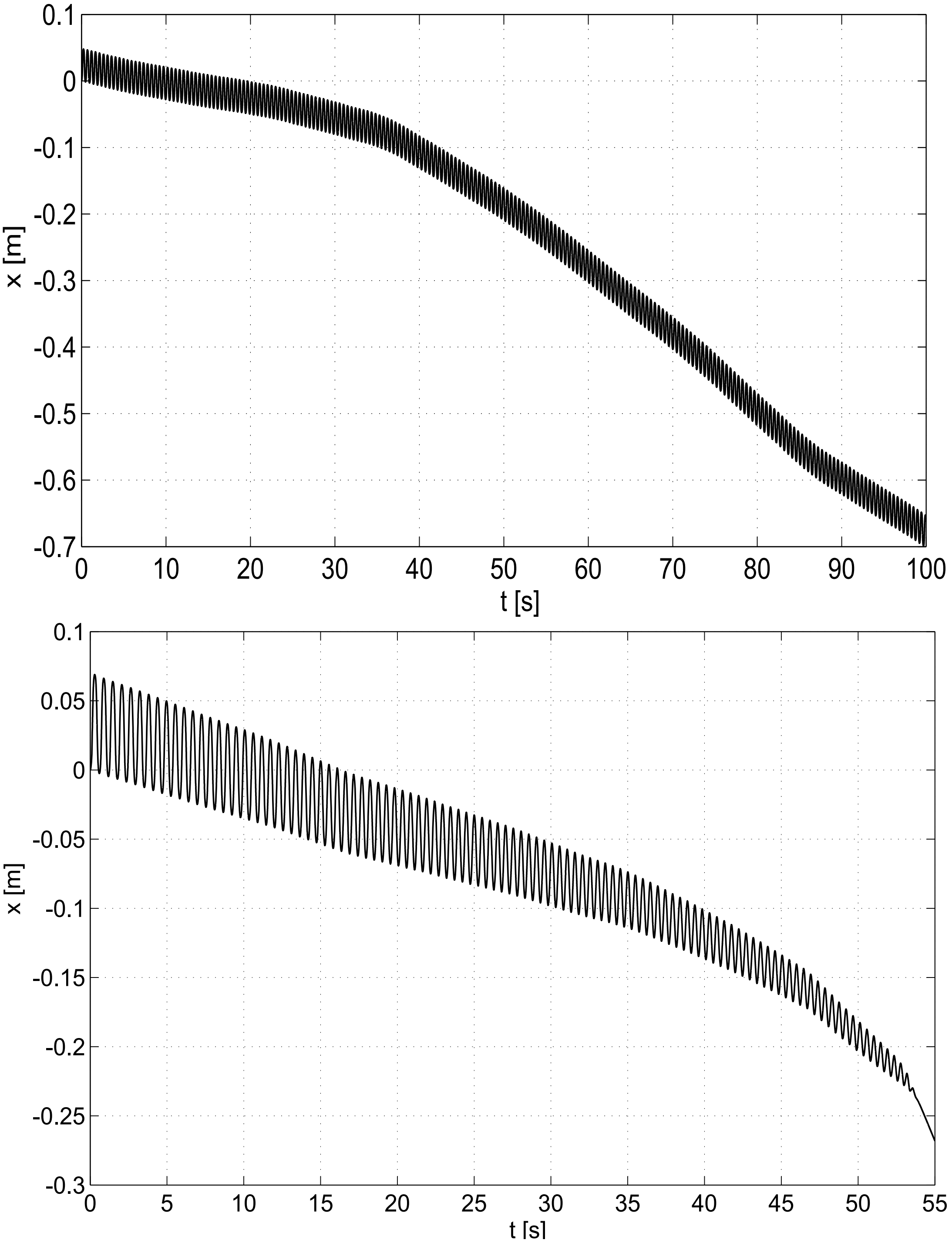


Figure 9.12: The position of the cart with controllers C_1 (upper) and with C_2 (lower), without noise, with very wide initial angle. Neither of them is stabilized because no friction is assumed and the model is approximated.

10

The Robust Fixed Point Transformations-based neural network controllers

In the previous chapter, a method can be seen how to combine RFPT with fuzzy logic controllers. In this chapter, the idea of improving SC-based controllers is further developed and extended to NN controllers. First, an improved RFPT and NN combined controller is introduced. Then, the author shows the steps of the application through the previously applied benchmark problem via the control of a cart-pendulum system. The NN is trained, built together with RFPT, and the control process is analyzed. The results show that RFPT can significantly increase the robustness of the NN controller.

After dealing with neural networks, in the next chapter, the applicability of RFPT in real life is investigated.

10.1 Introduction

Human recognition and control abilities far exceed those of complex intelligent control systems (e.g. robots). This has motivated scientist to analyze the human thinking to model nervous systems and use artificial neural networks in many areas (e.g. image precessing, signal processing, and control) [50]. There are two main advantages of using NNs: one of them is that NNs can well approximate the behavior of known, partially known, and unknown nonlinear systems. It can be useful since the controlled system is not always known to be able to control accurately. The other advantages are the learning and adaptation abilities of NNs. Since the controlled systems can change any time (due to external influence), it worths to use NNs instead of identifying the model every time it changes.

Since the 1980's neural networks have been widely used in the control field. In most of the applications they are used for two purposes: to identify a controlled system (see e.g. [95, 96, 97, 98, 99]), or to use as a controller (like in [12, 13, 14, 100, 101, 102]). To identify a controlled system, NNs can be very useful because the systems cannot be always modeled due to complexity or uncertainty issues. As controllers NNs can be very powerful if the controlled systems change during usage. The adaptation or learning algorithm (supervised or unsupervised) which is used to train the neural network can be applied also in on-line mode. Depending on which type of NN (feedforward or feedback) is used, various training algorithms can be applied. For example, for supervised learning of feedforward networks (e.g. multilayer perceptrons [51]), the well-known back-propagation algorithm [103] could be mentioned. For feedback NNs' unsupervised training Hopfield's method [104] is a proper choice. The difficulty is that the most popular on-line adaptations can be applied only if the system and the controller are both neural networks. Usually, the systems to be controlled are not neural networks, but can be approximated by them. In this case, because of the non-exactness of the models, the model approximation worsens the accuracy of the control.

To reduce the disadvantages caused by the approximation, some improvement on the control process is needed. One candidate for this can be the application of the Robust Fixed Point Transformations method. It proved to be advantageous in case of classical controllers and as it is shown in the previous chapter, in case of fuzzy controllers, as well. The author of this thesis suggests to combine RFPT also with NN controllers and shows in the following its advantages and the steps of the development and usage. The performance of the introduced extended controller is illustrated by an example. The presented Robust Fixed Point Transformations-based neural network controller is applied to the control of an inverted pendulum.

In the control process an approximate model of the cart is used which is not a neural network-based model, thus the on-line adaptation cannot be applied. The inaccuracy of the model causes difficulties in the control process especially when disturbances occur. The integration of RFPT can help to match the system and the model and thus, to

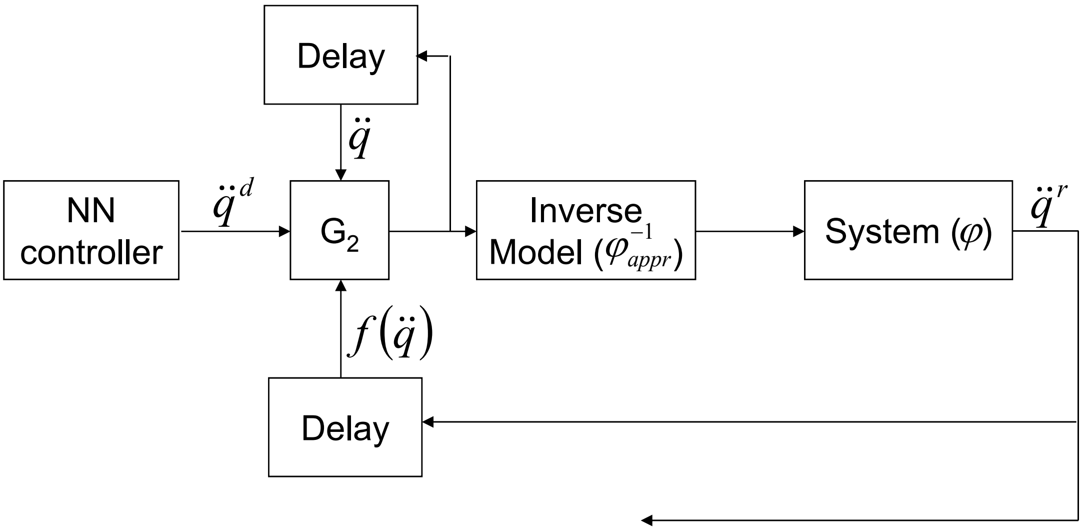


Figure 10.1: The block scheme of the RFPT-based Neural Network controller, where $\ddot{q} = [\ddot{x}_c, \ddot{\theta}]$.

overcome this problem. It helps to make the inaccurate neural network controller robust. The results show that a simple pre-trained (inaccurate) feedforward neural network controller can have difficulties when it faces disturbances, but RFPT can reduce the error caused by the disturbing forces.

10.2 The RFPT-based Neural Network Controller

Figure 10.1 shows the block diagram of the proposed controller. Assume that a desired response $\ddot{q}^d$ is prescribed for the system, which is determined by a neural network controller. From the desired response the proper control force has to be calculated for the system so that it responses in the prescribed way. The proper control force can be determined by an exact inverse model. Unfortunately, if the system is not known exactly, it is not possible to create an accurate inverse system: only an approximate inverse model can be of help. The approximate inverse model calculates the approximated control force for the system, which by this, results in a realized response (which differs from the desired one).

Since the controller's output is determined by assuming exact inverse system, by properly transforming the output of the controller (based on the approximate inversion), more accurate system response (closer to the desired response) can be gained. One of the possibilities for proper transformation is the function of the second type Robust Fixed Point Transformations (G_2). Because of the *tanh* function used in G_2, RFPT not just reduces the error caused by the model approximation, but makes the NN controller robust to the external disturbances.

In the previous chapter, it is shown how to combine fuzzy logic controllers with the first type of RFPT. In this chapter, similar efforts are taken with the second type of RFPT. The two combinations show a way how to ameliorate SC controllers with RFPT.

10.3 Simulation results

In this section, some illustrative simulation results are presented without limiting the generality of the new controller. The simulation task is balancing a pendulum ($\theta^N = \dot{\theta}^N = 0$) on the top of a cart of changing nominal position ($\dot{x}_c^N = 0, x_c^N \neq 0$). The details of the system is described in Section 3.5. The nominal position of the cart is described by a trapezoidal wave with $0.1\,m$ of amplitude and $12\,s$ of period. The simulations are made by the Matlab-Simulink package. In the examples the system parameters are set to $M = 1.5kg$, $m = 0.5kg$, $L_p = 0.1m$, $g = 9.8m/s^2$, and $b_{cp} = 0.1kg/s$.

The approximate model is assumed to have the same structure as the original cart-pendulum system. The model parameters are $\hat{M} = 1.2kg$, $\hat{m} = 0.8kg$, $\hat{L_p} = 0.09m$, $\hat{g} = 10m/s^2$ and $\hat{b}_{cp} = 0kg/s$. The initial values for the state variables are $x_c = 0$ and $\theta = \pi/10$.

The NN part of the proposed controller is a feedforward multilayer perceptron (see Subsection 2.3.2) supervised by a PD controller:

$$\ddot{q}^d = \ddot{q}^N + 2\lambda\left(\dot{q}^N - \dot{q}\right) + \lambda^2\left(q^N - q\right) \tag{10.1}$$

where $q = [x_c, \theta]$, q^N denotes the nominal trajectories for x_c and θ, and λ is a free parameter. The NNC has one hidden layer with ten neurons. The NNC has four inputs, the error in the angle ($e_\theta = \theta^N - \theta$) and the angular velocity ($\dot{e}_\theta = \dot{\theta}^N - \dot{\theta}$) of the pendulum, and the error in the position ($e_x = x_c^N - x_c$) and the velocity ($\dot{e}_x = \dot{x}_c^N - \dot{x}_c$) of the cart. From these values it calculates the desired second derivatives of the position ($\ddot{x}_c^d$) and angle ($\ddot{\theta}^d$) of the cart and the pendulum, respectively. It is trained by backpropagation. For the simulations the NN controller is generated by Matlab: the

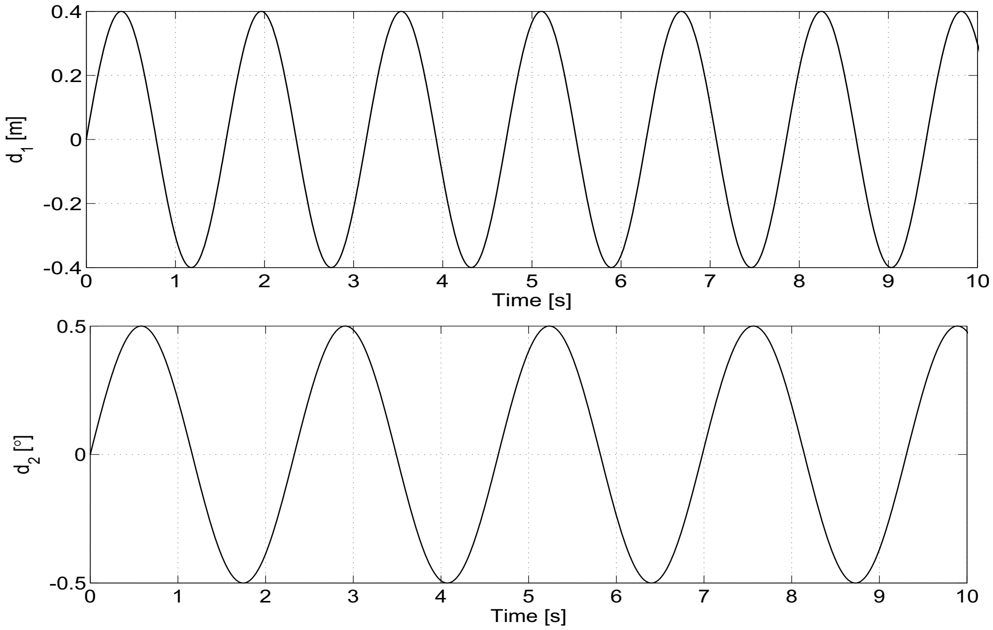


Figure 10.2: The disturbances effecting on $\ddot{x}_c$ ($T = 1.5708\,s$, upper) and effecting on $\ddot{\theta}$ ($T = 2.3271\,s$, lower).

training dataset is divided into two separate sets: one for training and one for testing. The ratio is set to 0.5. The number of epochs is set to 100, while the stopping error criteria (goal) is 0.1. The input vectors are $e_x = [-1; -0.8; ...; .0.8; 1] = e_\theta$ (e_x in m, e_θ in rad), and $\dot{e}_x = [-5; -4.5; ...; .4.5; 5] = e_\theta$ ($\dot{e}_x$ in m/s, e_θ in rad/s). After the training, a Simulink block is generated for the network by the *gensim* command.

During the simulations two controllers are compared: one without RFPT (C_1), and one with RFPT (C_2). Non disturbed control is also compared to the case with disturbances (which are sinusoidal waves, effecting $\ddot{x}_c$ and $\ddot{\theta}$, see Fig. 10.2).

In Fig. 10.3 four examples for the nominal and realized cart trajectories are illustrated. The results are a bit shifted because of the model approximation. When disturbances affect the system, the increased tracking error is visible only when RFPT is not used.

In Fig. 10.4 the tracking error in the positions of the cart is shown. Without disturbance the realized trajectories are a little bit shifted from the nominal one, otherwise they are very similar. The effect of the disturbance is observable only when RFPT is not applied. With disturbance the improvement seems obvious since without RFPT

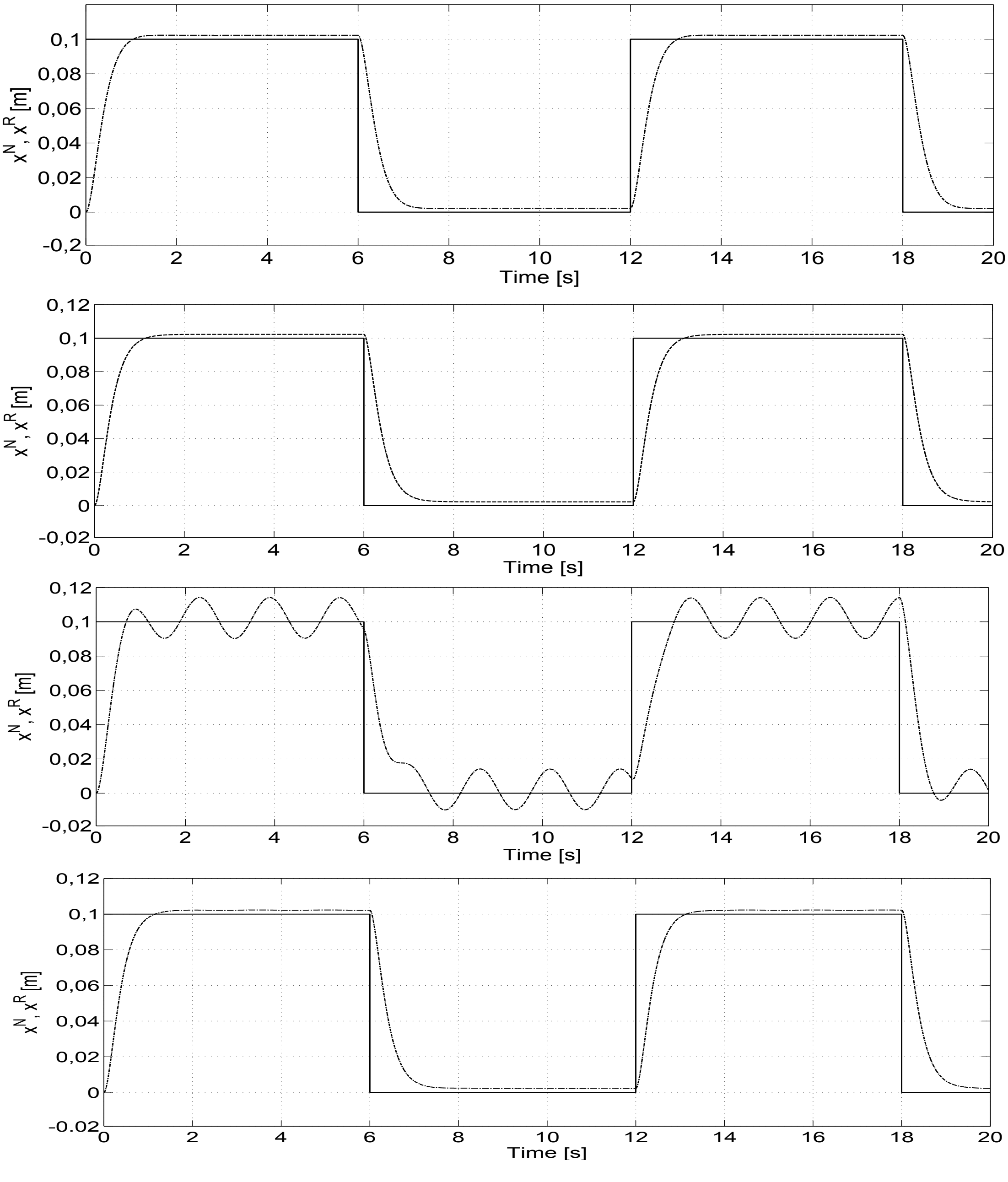


Figure 10.3: The nominal and realized trajectories of the cart with controllers C_1 and C_2: C_1 without disturbance (first); C_2 without disturbance (second); C_1 with disturbance (third); C_2 with disturbance (fourth). The trajectories are a bit shifted from the nominal ones. Without disturbance no significant difference can be seen. With disturbance the improvement of RFPT is obvious.

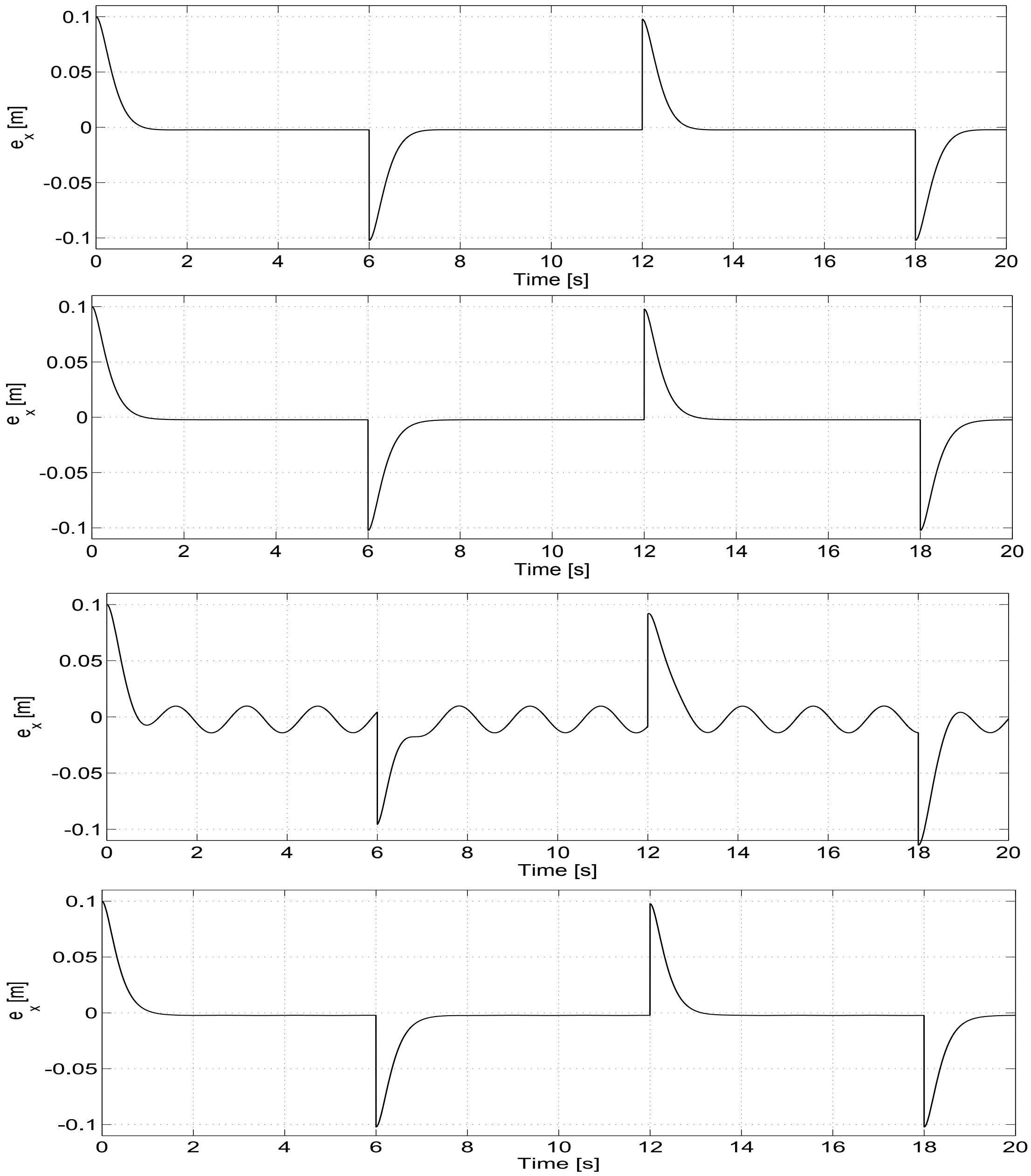


Figure 10.4: The error in the position of the cart with controllers C_1 and C_2: C_1 without disturbance (first); C_2 without disturbance (second); C_1 with disturbance (third); C_2 with disturbance (fourth). Without disturbance no significant difference can be seen, without RFPT the trajectories are a bit more shifted than with RFPT. With disturbance the improvement of RFPT is obvious.

there is a well observable fluctuation in the tracking error of x_c which depends on the disturbance.

In Fig. 10.5 the angles of the pendulum are illustrated for the above four examples. Without RFPT the errors can easily be seen and the disturbance strongly affect the angle. With RFPT the angles remain almost zero. The RFPT-based results are illustrated in Fig. 10.6 with zoom, to be able to determine the order and the shift of them.

10.4 Summary

Neural networks are widely used to investigate the behavior of unknown systems. In this paper, an extended neural network control scheme is introduced. The presented RFPT-based NNC is applied to balance a pendulum in the top of a cart of changing nominal position. The results show that if disturbance forces affect the system, the robustness of the controller is significantly increased by the use of RFPT and the tracking errors are decreased by orders of magnitude. Though in ideal environment the RFPT extension does not bring great improvement it cannot be assumed that real systems are not effected by external unknown forces.

The result considered to be new has been published in journal paper [J4].

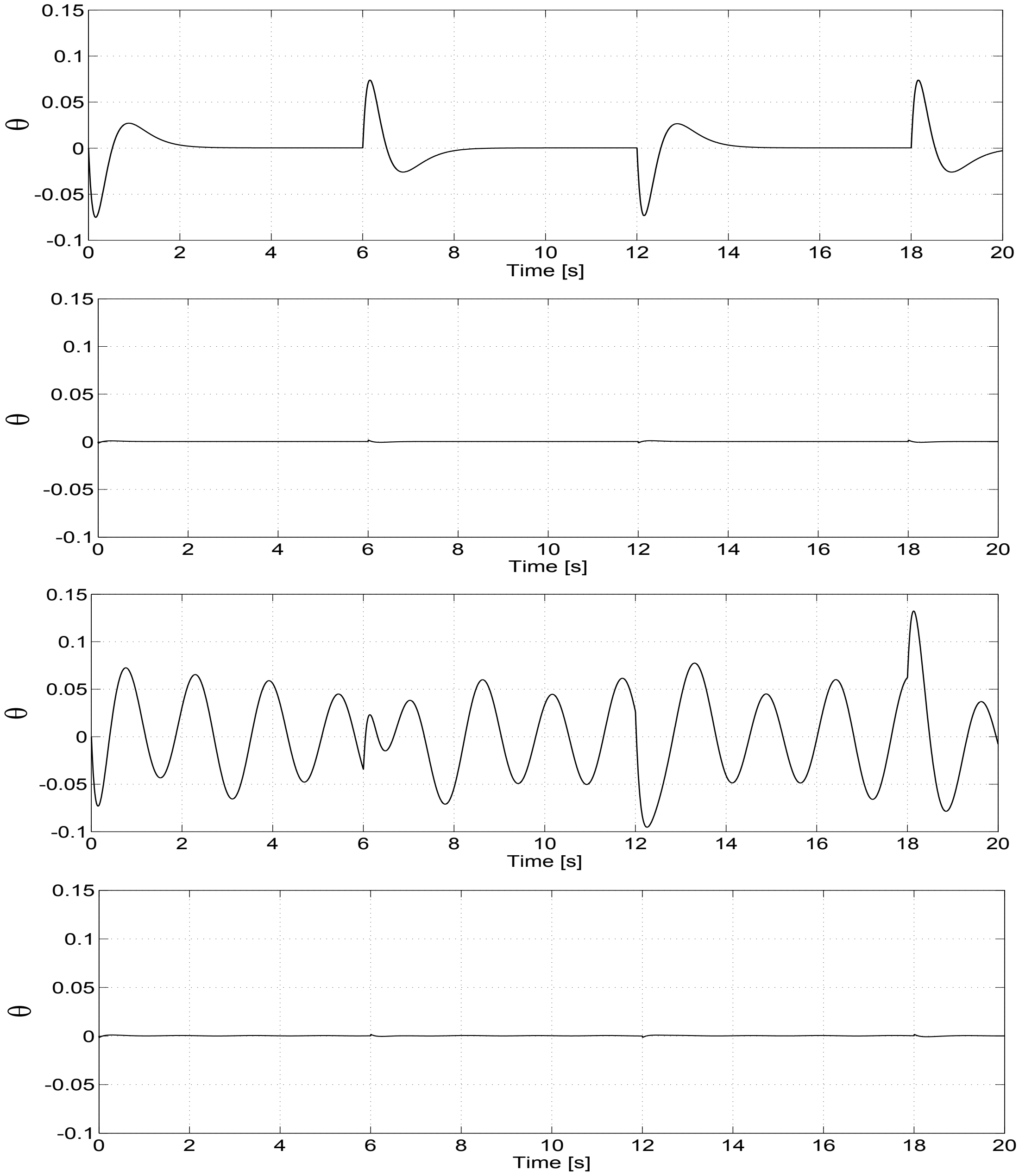


Figure 10.5: The angle of the pendulum with controllers C_1 and C_2: C_1 without disturbance (first); C_2 without disturbance (second); C_1 with disturbance (third); C_2 with disturbance (fourth). Without RFPT the pendulum traverses strongly because of the sudden position change of the cart. With RFPT the angle remain almost zero. The two RFPT-based figures are shown zoomed in Fig. 10.6.

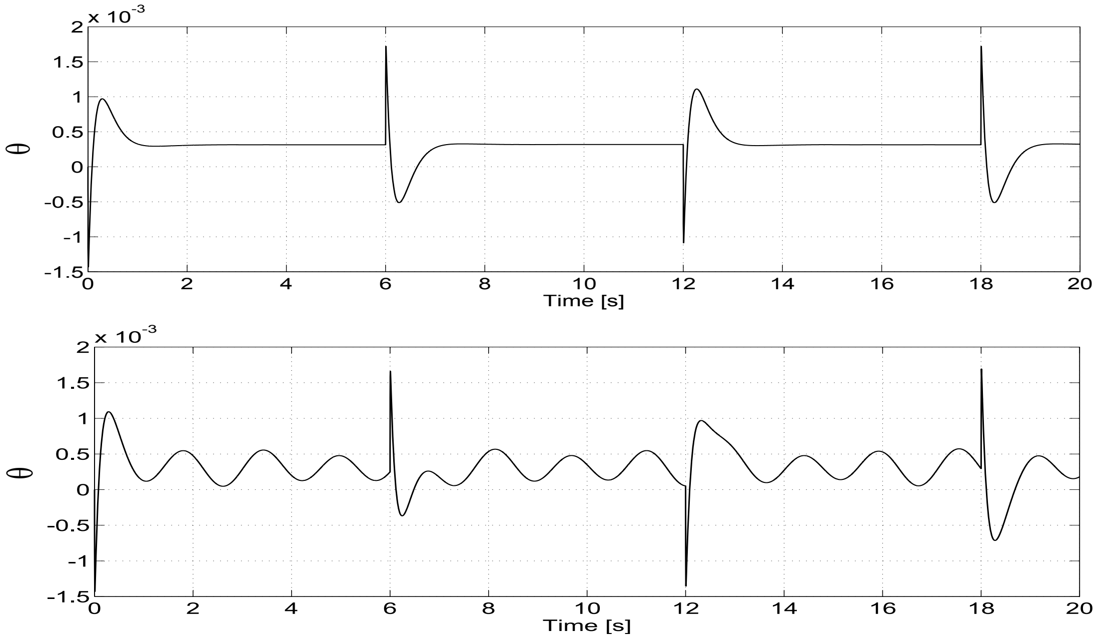


Figure 10.6: The zoomed angle of the pendulum with controller C_2: without disturbance (upper); with disturbance (lower). The improvement is around two orders of magnitude compared to the angles achieved with the traditional NN controller.

11

Emission control of exhaust fumes with Robust Fixed Point Transformations

In the previous two chapters, it is shown how the performance of soft computing-based, fuzzy and NN controllers can be improved with Robust Fixed Point Transformations. In this chapter, a special approach is investigated: RFPT is applied to a certain existing real life problem. The problem itself includes the control of a complex system, which has a direct effect on environmental pollution. It covers the traffic flow control considering the emission rate of exhaust fumes of freeway traffic.

As the first step, an approximate hydrodynamic model is analyzed (see Section 3.7) which uses only two parameters of the freeway traffic (vehicle density and traffic velocity). This type of model proved to be successful at other fields of engineering, e.g. Electronics and Nuclear Science (see e.g. [105, 106, 107, 108]) and also for traffic flow modeling (see e.g. [63, 109]). The model analysis includes the determination of the stationary solutions, followed by stability investigations.

As the second step, after determining the emission rate of exhaust fumes of freeway traffic, a new attribute is introduced which is strongly related to it and can more easily be measured by cheap loop detectors.

Finally, a control strategy is proposed to keep the emission rate of exhaust fumes at a desired level for the stationary solutions based on the introduced new attribute. Since the model approximation may lead to accuracy problems, the process is improved by the application of RFPT. The suggested technique can help to improve/optimize problems which are too complex to be modeled with enough precision to ensure appropriate control actions.

11.1 The basic control strategy in quasi-stationary approach

In Section 3.7, it is explained that there is a possibility to approximate the freeway traffic by a hydrodynamic model. The model needs only two properties of the traffic: the vehicle density ρ and the velocity v. According to this consumption, the system can be described by differential equations (3.16)-(3.23). In the following, the possible control strategies are analyzed based on the stability property of some given stationary solutions.

Assume that for a fixed additional input, i.e. control signal, $\hat{r}_2$ (see Fig. 3.12 and (3.16)-(3.23)) there exists a stationary solution system i.e. $\hat{\rho}_1$, $\hat{\rho}_2$, $\hat{\rho}_3$, $\hat{\rho}_4$, $\hat{v}_1$, $\hat{v}_2$, and $\hat{v}_3$, for which $\dot{\hat{\rho}}_1 = 0$, $\dot{\hat{\rho}}_2 = 0$, $\dot{\hat{\rho}}_3 = 0$, $\dot{\hat{\rho}}_4 = 0$, $\dot{\hat{v}}_1 = 0$, $\dot{\hat{v}}_2 = 0$, and $\dot{\hat{v}}_3 = 0$, but these solutions are unstable. Such a dynamic system requires very fast feedback signal in r_2 to stabilize the stationary solutions. In this case, the task is very difficult, and the successful practical implementation is dubious. But if the stationary solutions were stable, the control task could be approached in a far simpler way. In stable case, small steps in the control signal could result in small modifications of the controlled quantities.

The proposed control strategy is similar to that of some traditional approaches which are applied e.g. in Thermodynamics and Chemistry when the states of the thermal equilibrium are stable or at least metastable (i.e. they show stability at least against small perturbations). As an example some quasi-stationary thermodynamic processes can be mentioned where the state propagation is modeled as a sequence of stationary states like in [110, 111]. Similar example can be the stationary solutions of certain multiple-compartment process models of the human glucose-insulin system (e.g. [112, 113]). Their solutions all show stability that eases their control.

Based on the above, well investigated traditional results, if the stability of the quasy-stationary solutions of (3.16)-(3.23) could be found then a link could be established between the two approaches and the system model could be simply used for determining the necessary small steps in r_2. Thus, instead of applying fast dynamic feedback, a simple iterative controller (like the Robust Fixed Point Transformations-based controllers) could be used for the compensation of the modeling errors. In the next subsection the stationary solutions of the hydrodynamical model (explained in Section 3.7) are investigated and by that the necessary link is built.

11.1.1 The stationary solutions of the dynamic model

In Section 3.7, it is told that the quantities ρ_0, v_0 (consequently $q_0 = \rho_0 v_0$), v_4 and v_5 can be set as constant boundary conditions. If it is assumed that the control signal $\hat{r}_2$=constant, then (3.16)-(3.23) take the form of

$$\begin{aligned}
0 &= \frac{\hat{q}_0 - \hat{\rho}_1 \hat{v}_1}{L_{rs}\lambda_{ft}} && (11.1)\\
0 &= \frac{\hat{\rho}_1 \hat{v}_1 - \hat{\rho}_2 \hat{v}_2 + \hat{r}_2}{L_{rs}\lambda_{ft}} && (11.2)\\
0 &= \frac{\hat{\rho}_2 \hat{v}_2 - \hat{\rho}_3 \hat{v}_3}{L_{rs}\lambda_{ft}} && (11.3)\\
0 &= \frac{\hat{\rho}_3 \hat{v}_3 - \hat{\rho}_4 \hat{v}_4}{L_{rs}\lambda_{ft}} && (11.4)
\end{aligned}$$

$$\begin{aligned}
0 &= \frac{V(\hat{\rho}_1) - \hat{v}_1}{\tau} + \frac{\hat{v}_1(\hat{v}_0 - \hat{v}_2)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\hat{\rho}_2 - \hat{\rho}_0}{\hat{\rho}_1 + \kappa} && (11.5)\\
0 &= \frac{V(\hat{\rho}_2) - \hat{v}_2}{\tau} + \frac{\hat{v}_2(\hat{v}_1 - \hat{v}_3)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\hat{\rho}_3 - \hat{\rho}_1}{\hat{\rho}_2 + \kappa} - \frac{\delta}{L_{rs}} \frac{\hat{r}_2 \hat{v}_2}{\hat{\rho}_2 + \kappa} && (11.6)\\
0 &= \frac{V(\hat{\rho}_3) - \hat{v}_3}{\tau} + \frac{\hat{v}_3(\hat{v}_2 - \hat{v}_4)}{2L_{rs}} - \frac{\eta}{\tau 2L_{rs}} \frac{\hat{\rho}_4 - \hat{\rho}_2}{\hat{\rho}_3 + \kappa} && (11.7)
\end{aligned}$$

with the explicit equation for $\hat{\rho}_5$ as

$$\hat{\rho}_5 = \hat{\rho}_3 - \tfrac{\tau 2L_{rs}(\hat{\rho}_4 + \kappa)}{\eta} \times \left[\tfrac{V(\hat{\rho}_4) - \hat{v}_4}{\tau} + \tfrac{\hat{v}_4(\hat{v}_3 - \hat{v}_5)}{2L_{rs}}\right]. \qquad (11.8)$$

Due to nonlinearities in (11.1)-(11.7), the stationary solutions can be found in the following way. For reducing the complexity, a simple method can be proposed. The occurrence of $(\hat{\rho}_i + \kappa)$ in the denominators may cause division by zero in numerical algorithms. So in the first step such divisions are eliminated via multiplication in (11.1)-(11.6). It results

$$
\begin{aligned}
0 = f_1 &:= 2L_{rs}(\hat{\rho}_1\kappa)[V(\hat{\rho}_1) - \hat{v}_1] + \tau(\hat{\rho}_1 + \kappa)\hat{v}_1(\hat{v}_0 - \hat{v}_2) - \eta(\hat{\rho}_2 - \hat{\rho}_0) && (11.9)\\
0 = f_2 &:= 2L_{rs}(\hat{\rho}_2 + \kappa)[V(\hat{\rho}_2) - \hat{v}_2] + \tau(\hat{\rho}_2 + \kappa)\hat{v}_2(\hat{v}_1 - \hat{v}_3) - && (11.10)\\
&- \eta(\hat{\rho}_3 - \hat{\rho}_1) - 2\tau\delta\hat{r}_2\hat{v}_2 &&\\
0 = f_3 &:= 2L_{rs}(\hat{\rho}_3 + \kappa)[V(\hat{\rho}_3) - \hat{v}_3] + \tau(\hat{\rho}_3 + \kappa)\hat{v}_3(\hat{v}_2 - \hat{v}_4) - && (11.11)\\
&- \eta(\hat{\rho}_4 - \hat{\rho}_2) &&\\
0 = f_4 &:= \hat{q}_0 - \hat{\rho}_1\hat{v}_1 && (11.12)\\
0 = f_5 &:= \hat{\rho}_1\hat{v}_1 - \hat{\rho}_2\hat{v}_2 + \hat{r}_2 && (11.13)\\
0 = f_6 &:= \hat{\rho}_2\hat{v}_2 - \hat{\rho}_3\hat{v}_3 && (11.14)\\
0 = f_7 &:= \hat{\rho}_3\hat{v}_3 - \hat{\rho}_4\hat{v}_4 && (11.15)
\end{aligned}
$$

In the second step, (11.9)-(11.15) are solved by error optimization (minimization). The proposed optimization method is the generalized reduced gradient (GRG) method [114] because is has been proven to be a precise and accurate method for solving nonlinear programming problems. By solving the problem

$$\min f_1 : \forall i \; 2 \leq i \leq 7 \, f_i = 0 \& f_i > 0 \qquad (11.16)$$

where $50 \leq \hat{q}_0 \leq 800$ and $0 \leq \hat{r}_2 \leq 522$, it can be proven that there is dependence of the coefficients of the $\hat{r}_2$-based polynomial on $\hat{q}_0$ and the dependence can be well approximated by a third order polynomial. As a result, the stationary solutions can be well approximated in the investigated parameter range (with 3rd order two-variable polynomials according to $\hat{r}_2$ and $\hat{q}_0$).

In the last step, it is shown that the coefficients of the got third order polynomial depend on $\hat{q}_0$. The dependence is approximated also by third order polynomials.

Calculation tests can be made e.g. in MS Excel environment with the following steps:

1. Implement f_1 in the cells of worksheets, while $\forall i \; 2 \leq i \leq 7 \, f_i$ in Visual Basic macros.

2. Calculate the values $\forall i \, f_i$ with discrete modification of $\hat{r}_2$ from 0 to 522 (maximal load of the ramp).

3. Let $f_1 = 0$ the goal and $\forall i \; 2 \leq i \leq 7 \, f_i = 0$ the constraints of the solver.

4. Repeat the solver's calculations with different parameter settings, e.g. $\hat{q}_0$ =50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800.

The optimization for the last step can be done similarly.

Representative samples for the polynomial fitting are given in Figs. 11.1-11.3, where the model parameters are set as follows: the critical density is $\rho_{cr} = 26 \, \frac{1}{km}$ and $b = 2.5$; in the Papageorgiou model $\tau = 0.0087 \, h$, $\eta = 29.3505 \, \frac{km^2}{h}$, $\kappa = 20.77 \, \frac{1}{km}$, and $\delta = 0.05$; in the hydrodynamical model the number of lanes $\lambda_{ft} = 1$, and the length of the segments $L_{rs} = 0.5 \, km$; the velocity of the free flow is $v_{free} = 114 \, \frac{km}{h}$; the boundary conditions are set as $v_4 = v_5 = v_0 = 114 \, \frac{km}{h}$, and $\hat{r}_2$ is varied in $18 \, \frac{vehicle}{h}$ steps from 0 to 522; the calculations are repeated where $\hat{q}_0$ takes the {50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800} values.

In the next section, the emission factor of exhaust fumes is briefly considered and a new value is proposed for control purposes.

11.1.2 Introduction of the Emission Factor

For calculating the emission, it is assumed that at high velocities the most significant dissipative factor is the drag force generated by the eddying air. The drag force is proportional to the square of the velocity: $F = Cv^2$ in which coefficient C depends on the particular vehicles. At velocity v the power consumption of this drag force is $Fv = Cv^3$ which roughly determines its fuel consumption. On a road segment with length L_{rs} and at vehicle density ρ, the number of cars on the road segment can be determined as $L_{rs}\rho$. The vehicles on the given road segment produce $L_{rs}\overline{C}\rho v^3$ power consumption which roughly determines the emission rate of exhaust fumes on this segment ($\overline{C}$ denotes some average for the various vehicles present on the segment). It is very difficult to obtain information on $\overline{C}$, however, the emission for any car still strongly depends on the factor

$$E_f := \rho v^3. \tag{11.17}$$

E_f can be measured easily by cheap and simple inductive loop detectors (see e.g. [115, 116]). On this reason, (11.17) is referred to as the emission factor. In case of stationary flow, E_f can be expressed in road segment 3 as

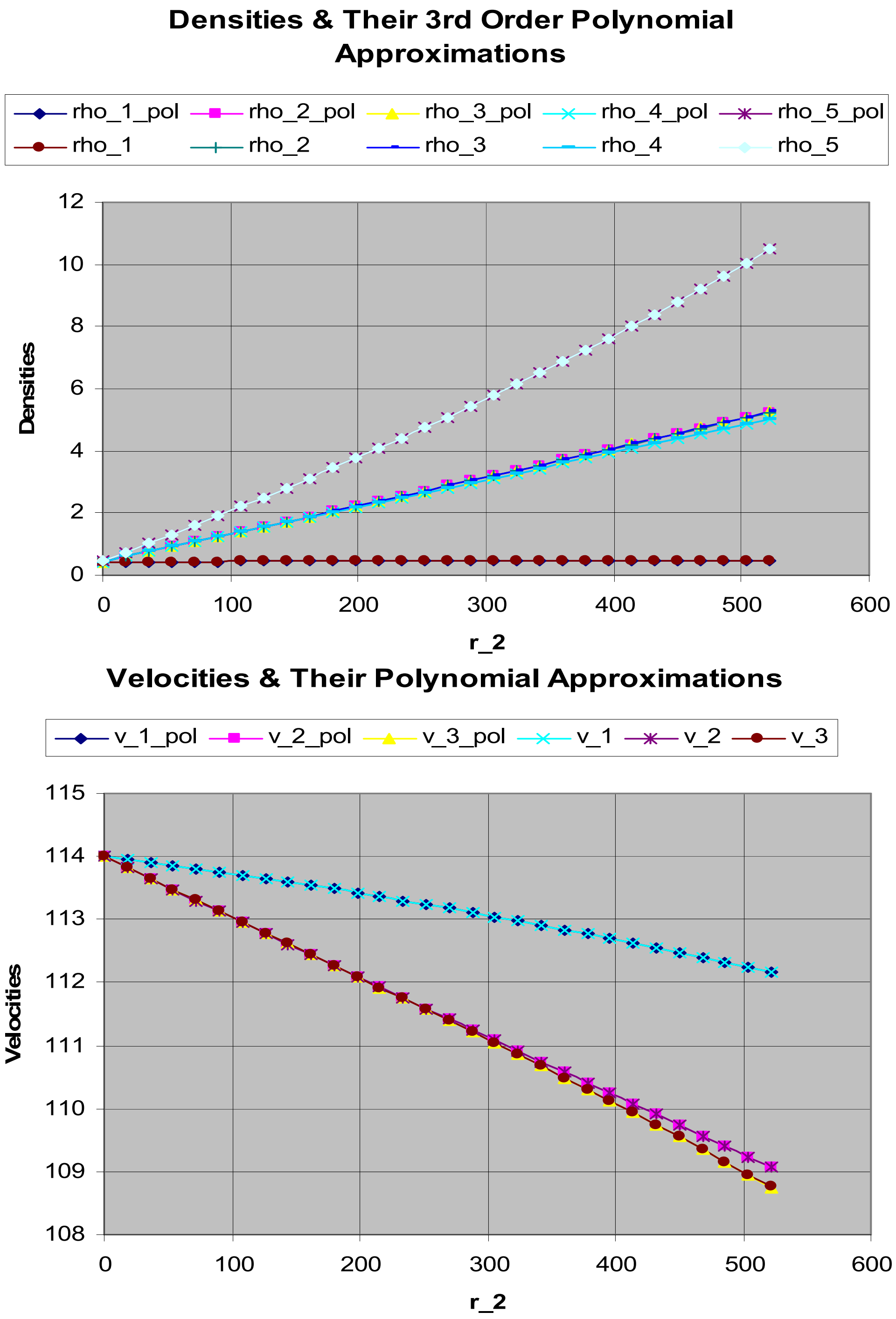


Figure 11.1: Fitted third order polynomials (rho_i_pol, upper; v_i_pol, lower) for the parameters(rho_i and v_i, respectively) for $\hat{q}_0 = 50$ (where $i \in \mathbb{N}$).

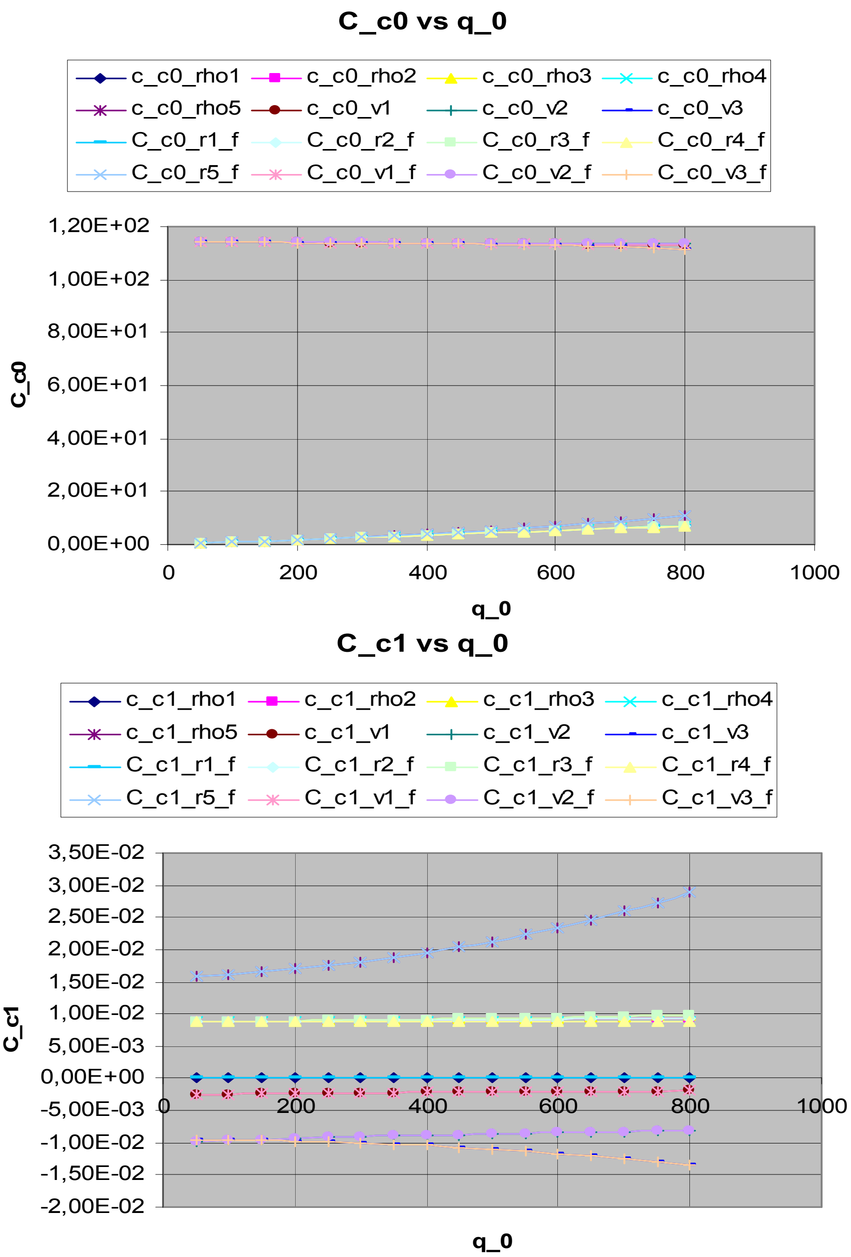


Figure 11.2: The dependence of the first two coefficients of the $\hat{r}_2$-based polynomial (*c_c0_var* and *c_c1_var*, where *var* may represents rho1, rho2, rho3, rho3, rho5, v1, v2, or v3) on $\hat{q}_0$ and the third order polynomial approximation (*C_c0_var_f* and *C_c1_var_f*) of this dependence.

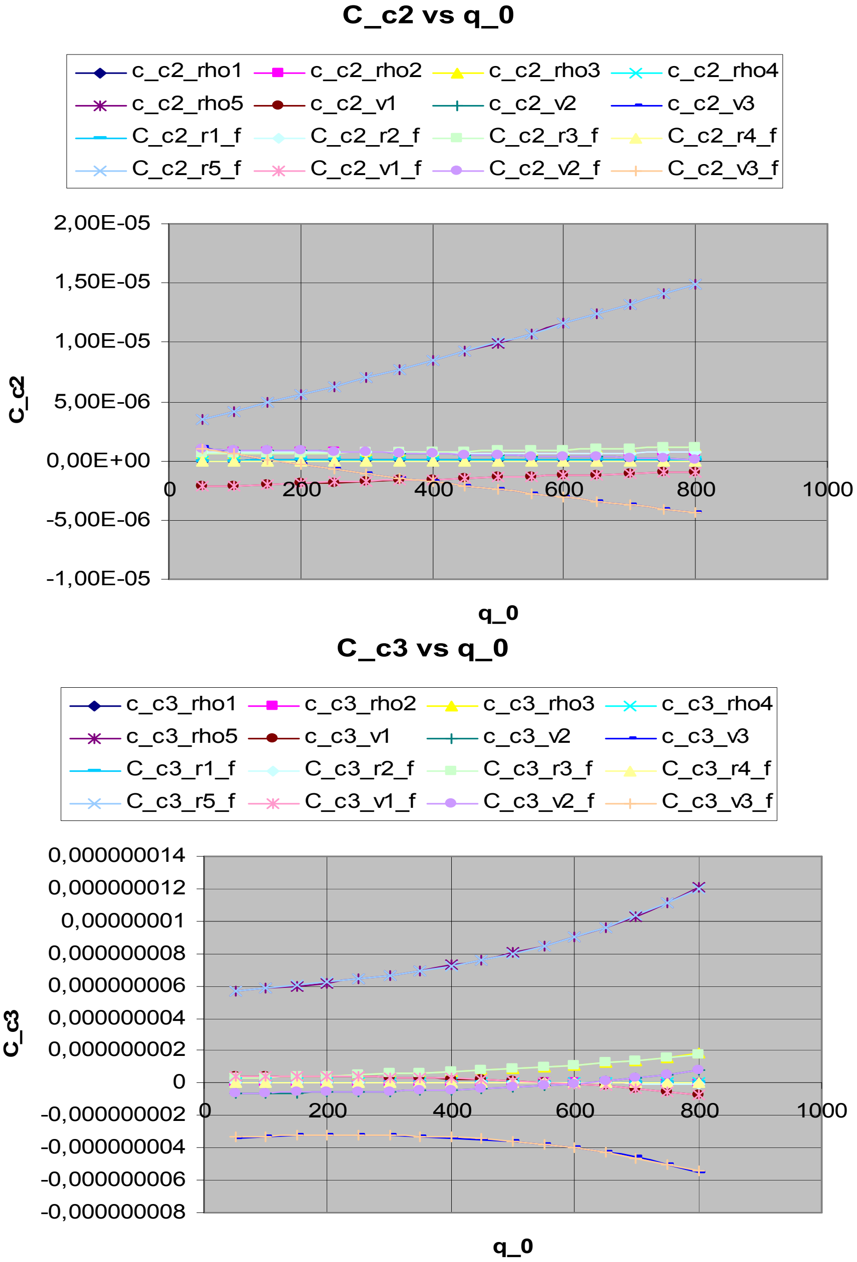


Figure 11.3: The dependence of the first two coefficients of the $\hat{r}_2$-based polynomial (c_c2_var and c_c3_var, where var may represents rho1, rho2, rho3, rho3, rho5, v1, v2, or v3) on $\hat{q}_0$ and the third order polynomial approximation ($C\_c2\_var\_f$ and $C\_c3\_var\_f$) of this dependence.

$$E_f = (\hat{q}_0 + \hat{r}_2)v_3^2 \tag{11.18}$$

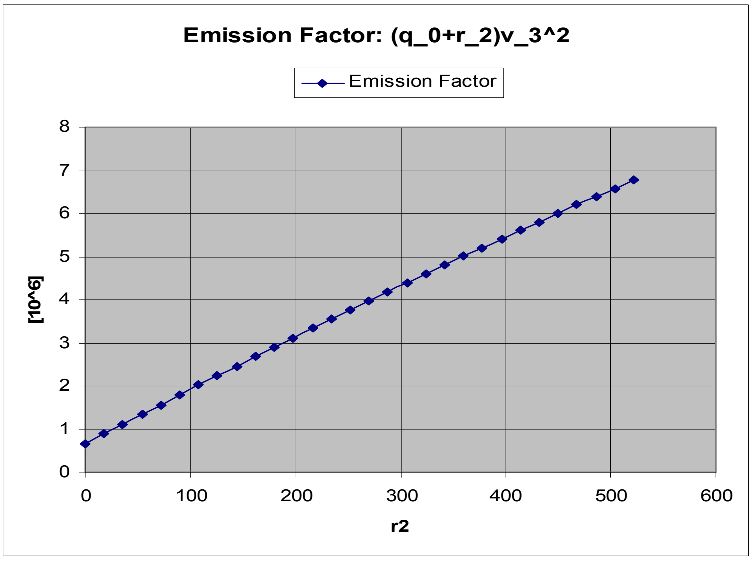
since $\hat{\rho}_3\hat{v}_3 = \hat{q}_0 + \hat{r}_2$. Independently of the actual (unknown) value of $\overline{C}$, this factor must be decreased if the contamination of air is too high, or it can be increased if the actual concentration of the exhaust fumes in the air is under some prescribed threshold. On this reason direct polynomial fitting for E_f can be made, too (see Fig. 11.4).

To sum up, the stationary behavior of the system can be well approximated in the given parameter range by a few simple matrices (containing the coefficients of the polynomial fitting). On the basis of the above calculations, it can be observed that the emission factor E_f is monotonous strongly increasing function of $\hat{r}_2$ in each case. Therefore in the possession of the actual $\hat{q}_0$ value, by decreasing $\hat{r}_2$ step-by-step from a big initial value, the function of E_f can numerically be inverted. With the inversion the appropriate $\hat{r}_2$ can numerically be determined for a desired E_f. On the other hand, due to the drastic increase in E_f (depending on $\hat{\rho}_3$ and $\hat{v}_3$), it can be expected that relatively small model parameter errors may lead to drastic estimation error in E_f (if it is wished to be estimated on the basis of the model). Therefore simultaneous measurement of $\hat{\rho}_3$ and $\hat{v}_3$ is desirable.

For the use of the stationary model the stability of the stationary states is required. This problem is discussed in the next subsection.

11.1.3 Formal analysis of the stability of the stationary solutions

The fundamental approach for the stability analysis is based on perturbation calculus according to which first order linear differential equations are derived for the little perturbations of the stationary states. In this approach, the third or higher order terms in the perturbations are simply neglected. Accordingly, $\{\rho_i := \hat{\rho}_i + \epsilon\rho_i | i = 1,2,3,4\}$, $\{v_i := \hat{v}_i + \epsilon v_i | i = 1,2,3\}$, $\{\dot{\rho}_i := \epsilon\dot{\rho}_i | i = 1,2,3,4\}$, $\{\dot{v}_i := \epsilon\dot{v}_i | i = 1,2,3\}$, and $\hat{q}_0$ =constant, $\hat{\rho}_0$ =constant, $\hat{v}_0$ =constant, $\hat{v}_4$ =constant, $\hat{v}_5$ =constant, and finally $\hat{r}_2$ =constant. By neglecting the higher order terms and taking into account the already known information on the stationary states, the following set of linear equations is obtained for the time-dependence of the perturbations:



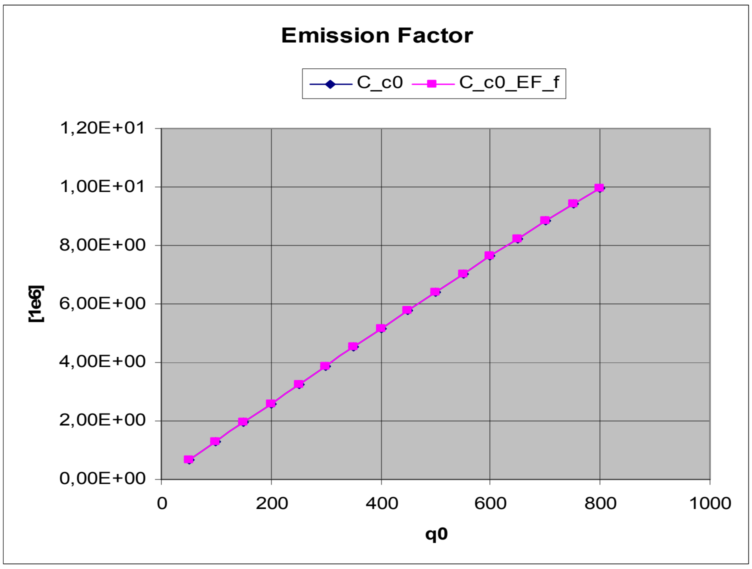


Figure 11.4: Fitted third order polynomial for the emission factor for $\hat{q}_0 = 50$ (upper) and the dependence of the first coefficient of the ($\hat{r}_2$-based) polynomial (C_c0) on $\hat{q}_0$ and its third order polynomial approximation (C_c0_EF_f; lower).

$$\epsilon\dot{\rho}_1 \approx \frac{\hat{q}_0 - (\hat{\rho}_1 + \epsilon\rho_1)(\hat{v}_1 + \epsilon v_1)}{L_{rs}\lambda_{ft}} \approx \frac{\hat{q}_0 - \hat{\rho}_1\hat{v}_1 - \hat{v}_1\epsilon\rho_1 - \hat{\rho}_1\epsilon v_1}{L_{rs}\lambda_{ft}} \approx -\frac{\hat{v}_1\epsilon\rho_1 - \hat{\rho}_1\epsilon v_1}{L_{rs}\lambda_{ft}} \tag{11.19}$$

In similar manner:

$$\epsilon\dot{\rho}_2 \approx \frac{\hat{v}_1\epsilon\rho_1 + \hat{\rho}_1\epsilon v_1 - \hat{v}_2\epsilon\rho_2 - \hat{\rho}_2\epsilon v_2}{L_{rs}\lambda_{ft}}, \tag{11.20}$$

$$\epsilon\dot{\rho}_3 \approx \frac{\hat{v}_2\epsilon\rho_2 + \hat{\rho}_2\epsilon v_2 - \hat{v}_3\epsilon\rho_3 - \hat{\rho}_3\epsilon v_3}{L_{rs}\lambda_{ft}}. \tag{11.21}$$

For calculating the perturbations of the velocity components, the following approximations are needed: $\frac{1}{x+\epsilon x} \approx \frac{1}{x} - \frac{\epsilon x}{x^2}$ and $V(\hat{\rho}_1 + \epsilon\rho_1) \approx V(\hat{\rho}_1) + V'(\hat{\rho}_1)\epsilon\rho_1$. With these approximations it can be obtained that

$$\begin{gathered}\epsilon\dot{v}_1 \approx \tfrac{V(\hat{\rho}_1)+V'(\hat{\rho}_1)\epsilon\rho_1-\hat{v}_1-\epsilon v_1}{\tau} + \tfrac{(\hat{v}_1+\epsilon v_1)(\hat{v}_0-\hat{v}_2-\epsilon v_2)}{2L_{rs}} - \\ -\tfrac{\eta}{\tau 2L_{rs}}\left[\tfrac{1}{\hat{\rho}_1+\kappa} - \tfrac{\epsilon\rho_1}{(\hat{\rho}_1+\kappa)^2}\right](\hat{\rho}_2 + \epsilon\rho_2 - \hat{\rho}_0)\end{gathered} \tag{11.22}$$

that contains the following 0th and 1st order terms:

$$\begin{gathered}\epsilon\dot{v}_1 \approx \tfrac{V(\hat{\rho}_1)-\hat{v}_1}{\tau} + \tfrac{\hat{v}_1(\hat{v}_0-\hat{v}_2)}{2L_{rs}} - \tfrac{\eta}{\tau 2L_{rs}}\tfrac{\hat{\rho}_2-\hat{\rho}_0}{\hat{\rho}_1+\kappa} + \\ +\tfrac{V'(\hat{\rho}_1)\epsilon\rho_1-\epsilon v_1}{\tau} + \tfrac{-\hat{v}_1\epsilon v_2+(\hat{v}_0-\hat{v}_2)\epsilon v_1}{2L_{rs}} - \tfrac{\eta}{\tau 2L_{rs}}\left[\tfrac{\epsilon\rho_2}{\hat{\rho}_1+\kappa} - \tfrac{(\hat{\rho}_2-\hat{\rho}_0)\epsilon\rho_1}{(\hat{\rho}_1+\kappa)^2}\right].\end{gathered} \tag{11.23}$$

By utilizing the stationary equations and selecting the coefficients of the 0-2$^{\text{nd}}$ order terms in the perturbations, it is obtained that

$$\begin{gathered}\epsilon\dot{v}_1 \approx \left[\tfrac{V'(\hat{\rho}_1)}{\tau} + \tfrac{\eta(\hat{\rho}_2-\hat{\rho}_0)}{\tau 2L_{rs}(\hat{\rho}_1+\kappa)^2}\right]\epsilon\rho_1 - \\ -\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_1+\kappa)}\epsilon\rho_2 + \left[-\tfrac{1}{\tau} + \tfrac{\hat{v}_0-\hat{v}_2}{2L_{rs}}\right]\epsilon v_1 - \tfrac{\hat{v}_1}{2L_{rs}}\epsilon v_2.\end{gathered} \tag{11.24}$$

Similar considerations can be applied for $\epsilon\dot{v}_2$ resulting in

$$\begin{gathered}\epsilon\dot{v}_2 \approx \left[\tfrac{V'(\hat{\rho}_2)}{\tau} + \tfrac{\eta(\hat{\rho}_3-\hat{\rho}_1)}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)^2} + \tfrac{\epsilon\hat{r}_2\hat{v}_2}{L_{rs}(\hat{\rho}_2+\kappa)^2}\right]\epsilon\rho_2 + \left[\tfrac{-1}{\tau} + \tfrac{\hat{v}_1-\hat{v}_3}{2L_{rs}} - \tfrac{\epsilon\hat{r}_2}{L_{rs}(\hat{\rho}_2+\kappa)}\right]\epsilon v_2 + \\ +\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)}\epsilon\rho_1 - \tfrac{\hat{v}_2}{2L_{rs}}\epsilon v_3 - \tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)}\epsilon\rho_3 + \tfrac{\hat{v}_2}{2L_{rs}}\epsilon v_1.\end{gathered} \tag{11.25}$$

Finally, for $\epsilon\dot{v}_3$ it is yielded that

$$\begin{gathered}\epsilon\dot{v}_3 \approx \left[\tfrac{V'(\hat{\rho}_3)}{\tau} + \tfrac{\eta(\hat{\rho}_4-\hat{\rho}_2)}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)^2}\right]\epsilon\rho_3 + \left[\tfrac{-1}{\tau} + \tfrac{\hat{v}_2-\hat{v}_4}{2L_{rs}}\right]\epsilon v_3 + \\ +\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)}\epsilon\rho_2 - \tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)}\epsilon\rho_4 + \tfrac{\hat{v}_3}{2L_{rs}}\epsilon v_2\end{gathered} \tag{11.26}$$

By the use of these calculations, a simple matrix equation form $\dot{x} = Ax$ can obtained for the small state perturbations in which $x := (\epsilon\rho_1, \epsilon\rho_2, \epsilon\rho_3, \epsilon\rho_4, \epsilon v_1, \epsilon v_2, \epsilon v_3)^T$, and

$$A = \begin{pmatrix} A_{1,1} & 0 & 0 & 0 & A_{1,5} & 0 & 0 \\ A_{2,1} & A_{2,2} & 0 & 0 & A_{2,5} & A_{2,6} & 0 \\ 0 & A_{3,2} & A_{3,3} & 0 & 0 & A_{3,6} & A_{3,7} \\ 0 & 0 & A_{4,3} & A_{4,4} & 0 & 0 & A_{4,7} \\ A_{5,1} & A_{5,2} & 0 & 0 & A_{5,5} & A_{5,6} & 0 \\ A_{6,1} & A_{6,2} & A_{6,3} & 0 & A_{6,5} & A_{6,6} & A_{6,7} \\ 0 & A_{7,2} & A_{7,3} & A_{7,4} & 0 & A_{7,6} & A_{7,7} \end{pmatrix} \tag{11.27}$$

The identically non-zero terms are individually marked in A. The appropriate matrix elements are obtained from the above perturbation calculus as

$$\begin{aligned}
&A_{1,1} = \tfrac{-\hat{v}_1}{L_{rs}\lambda_{ft}};\ A_{1,5} = \tfrac{-\hat{\rho}_1}{L_{rs}\lambda_{ft}};\ A_{2,5} = \tfrac{\hat{\rho}_1}{L_{rs}\lambda_{ft}}; A_{2,1} = \tfrac{\hat{v}_1}{L_{rs}\lambda_{ft}};\ A_{2,2} = \tfrac{-\hat{v}_2}{L_{rs}\lambda_{ft}};\\
&A_{3,2} = \tfrac{\hat{v}_2}{L_{rs}\lambda_{ft}};\ A_{3,3} = \tfrac{-\hat{v}_3}{L_{rs}\lambda_{ft}};\ A_{3,6} = \tfrac{\hat{\rho}_2}{L_{rs}\lambda_{ft}}; A_{3,7} = \tfrac{-\hat{\rho}_3}{L_{rs}\lambda_{ft}};\\
&A_{4,3} = \tfrac{\hat{v}_3}{L_{rs}\lambda_{ft}};\ A_{4,4} = \tfrac{-\hat{v}_4}{L_{rs}\lambda_{ft}};\ A_{4,7} = \tfrac{\hat{\rho}_3}{L_{rs}\lambda_{ft}}; A_{5,1} = \left[\tfrac{V'(\hat{\rho}_1)}{\tau} + \tfrac{\eta(\hat{\rho}_2-\hat{\rho}_0)}{\tau 2L_{rs}(\hat{\rho}_1+\kappa)^2}\right];\\
&A_{5,2} = -\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_1+\kappa)}; A_{5,5} = \left[-\tfrac{1}{\tau} + \tfrac{\hat{v}_0-\hat{v}_2}{2L_{rs}}\right];\ A_{5,6} = -\tfrac{\hat{v}_1}{2L_{rs}};\\
&A_{6,2} = \left[\tfrac{V'(\hat{\rho}_2)}{\tau} + \tfrac{\eta(\hat{\rho}_3-\hat{\rho}_1)}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)^2} + \tfrac{\epsilon\hat{r}_2\hat{v}_2}{L_{rs}(\hat{\rho}_2+\kappa)^2}\right] A_{6,6} = \left[\tfrac{-1}{\tau} + \tfrac{\hat{v}_1-\hat{v}_3}{2L_{rs}} - \tfrac{\epsilon\hat{r}_2}{L_{rs}(\hat{\rho}_2+\kappa)}\right];\\
&A_{6,1} = \tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)};\ A_{6,7} = -\tfrac{\hat{v}_2}{2L_{rs}}; A_{6,3} = -\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_2+\kappa)};\ A_{6,5} = \tfrac{\hat{v}_2}{2L_{rs}};\\
&A_{7,3} = \left[\tfrac{V'(\hat{\rho}_3)}{\tau} + \tfrac{\eta(\hat{\rho}_4-\hat{\rho}_2)}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)^2}\right]; A_{7,7} = \left[\tfrac{-1}{\tau} + \tfrac{\hat{v}_2-\hat{v}_4}{2L_{rs}}\right];\\
&A_{7,2} = \tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)};\ A_{7,4} = -\tfrac{\eta}{\tau 2L_{rs}(\hat{\rho}_3+\kappa)}; A_{7,6} = \tfrac{\hat{v}_3}{2L_{rs}}.
\end{aligned} \tag{11.28}$$

From the control theory of linear, parameter-invariant systems it is known that the satisfactory and necessary condition of stability is that the eigenvalues of matrix A must have only negative real parts (see [117]). Therefore, by the use of the polynomial fitting of the stationary states of (3.16)-(3.23), the spectrum of A is determined numerically by varying $\hat{q}_0$ in $50/h$ units and gently varying $\hat{r}_2 \in [0, 522]\, vehicle/h$. All the solutions are found to be stable, though, they contain damped fluctuations. Therefore, the simple control approach based on the automatic relaxation of the perturbations of the quasi-stationary states is adaptable. Since in the analyzed problem a rough model is used, the controller may also need iterative adaptive corrections for which the author suggests the use of Robust Fixed Point Transformations.

11.2 Simulation results

The effectiveness of the proposed control strategy is investigated via simulations in Scilab-SCICOS environment. In the examples, freeway traffic is controlled without (C_1) and with RFPT (C_2). The aim of the control is that the emission factor tracks a nominal trajectory which in all of the presented examples is a sinusoidal wave.

In the illustrative examples, without limiting the generality, the following parameter values are used: for sampling time $\Delta t_{sampling} = 0.028\,h \approx 100.8\,s$, for the free parameters of the second type of RFPT (see Section 4.4) $K = -10^{10}$, $A = 5 \times 10^{-12}$, and $B = 1$, and for the maximum step size of the integrator $\Delta t_{sampling}/50$ are chosen.

For the control of the emission factor at road segment 3 (see Fig. 3.12), the 3$^{\mathrm{rd}}$ order polynomial fitting of E_f is directly calculated. Utilizing the fact that E_f is a monotonously increasing function of $\hat{r}_2$ (for arbitrary positive $\hat{q}_0$), a simple inverse function can be utilized to find a model-based $\hat{r}^d_{2_{appr}}$ value for a prescribed $\hat{E}_f^{Nom} \equiv \hat{E}_f^d$ emission factor. The controller without RFPT (C_1) directly introduces this value to the inverse model. The RFPT-based controller (C_2) transforms (improves) the gained value to calculate a better input for the approximate model. The structural scheme of the controller is shown in Fig. 11.5.

Numerous simulations have been made for exact and approximate inverse models (RFPT is needed in both cases because the polynomial fitting causes approximation). In

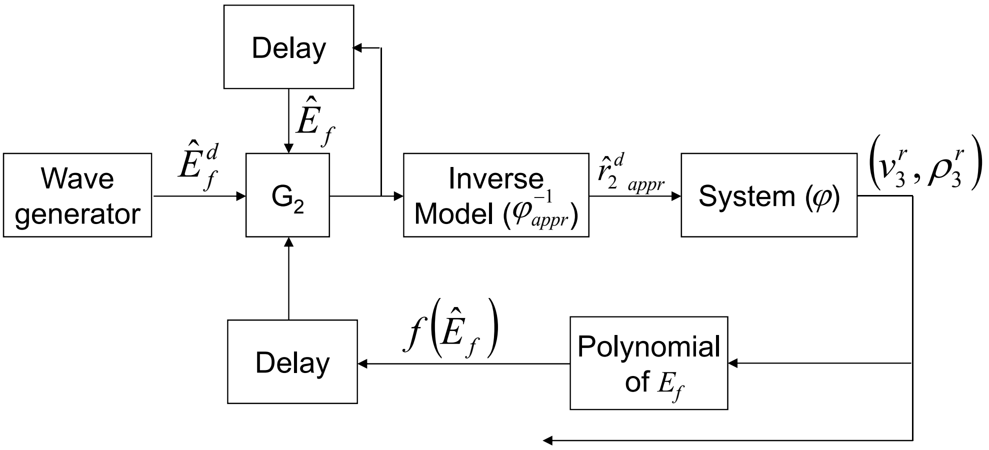


Figure 11.5: The block scheme of the RFPT-based control of E_f.

the analyses the used approximate model had the same structure as the inverse system, however with different parameter settings. The approximate parameters (marked by symbol $\sim$) are set as follows: $\tilde{v}_{free} = 1.20 v_{free}$, $\tilde{b} = 1.2b$, $\tilde{L}_{rs} = L_{rs}$, $\tilde{\rho}_{cr} = 1.2\rho_{cr}$, $\tilde{\tau} = 1.2\tau$, $\tilde{\eta} = 1.2\eta$, $\tilde{\kappa} = 1.2\kappa$, $\tilde{\delta} = 1.2\delta$, and $\tilde{\lambda}_{ft} = \lambda_{ft}$. The not enumerated parameters' values are also increased by 20%.

In the first example, in the simulations $\hat{q}_0$ is varied in drastic steps while E_f^d varies continuously (see Fig. 11.6) and the cycle time of the controller Δt_{Cycle} is set to be very big ($\Delta t_{Cycle} \approx 100\,s$). The given situation has been investigated using exact model parameter settings and both controllers C_1 and C_2, then using approximate model parameters and controller C_2. Figure 11.7 shows the tracking errors achieved in the three different situations. The first figure (using controller C_1) reveals that the fitted stationary approximation is in harmony with the output of the dynamic model, but the sign of the tracking error is identical in the great majority of the simulation time (the approximation is a little bit shifted from the nominal values). With controller C_2 the error oscillates always around zero, so the RFPT-based controller achieves good tracking of the emission factor E_f. Although, the model approximation increases the tracking error, but not significantly. Figure 11.8 shows the variation of r_2 during simulations.

In the next example, in the simulations both $\hat{q}_0$ and E_f^d vary continuously (see Fig. 11.9) and the approximate model parameters are used during all the simulations. In the practice in urban traffic the available time for crossing a street is about $10\,s$, so better accuracy can be expected with smaller sampling time than that of the previous example. The first two simulations are made with high sampling time, but in the third example Δt_{Cycle} is decreased to $10\,s$. The tracking error of the emission factor is shown in Fig. 11.10. The simulations show that with controller C_1 the error remains shifted (like in the previous example) and gets ten times bigger because of the parameter approximations. Thus, E_f drastically depends on the model parameters. On the other hand, with controller C_2 the error fluctuates around zero and its order of magnitude does not increase. Further, if low sample time is applied, the error is reduced significantly. The variation of r_2 is illustrated in Fig. 11.11.

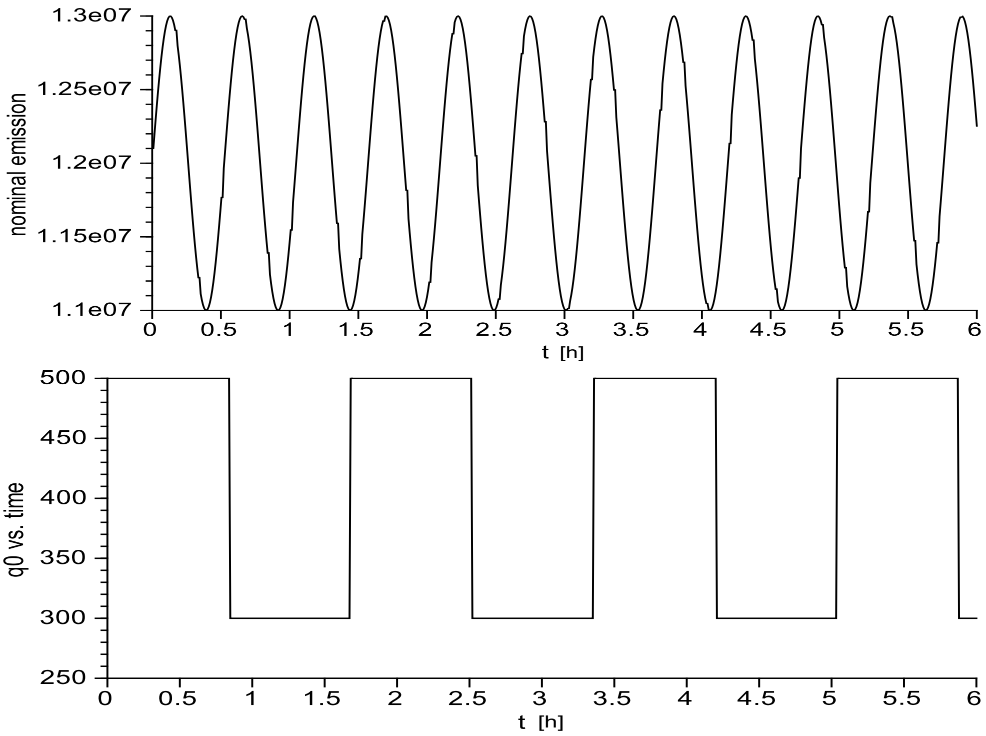


Figure 11.6: Ex1.: The nominal emission factor (upper; in km^2/h^3 units) and the variation of $\hat{q}_0$ (lower). In the first example, $\hat{q}_0$ is varied in drastic steps while E_f^d varies continuously.

11.3 Summary

In this chapter, a possible application of Robust Fixed Point Transformations is proposed. The task is to solve the control of the emission rate of exhaust fumes of freeway traffic based on a given approximate hydrodynamic traffic model. First, a link has been established between different successful applications of such models and the current problem: a numerical method is introduced for determining the stationary solutions of the system and then the stability of the solutions is shown. Finally, a simple RFPT-based control strategy is presented based on an introduced attribute (related to the emission rate), which successfully can handle the system even in case of rough model approximation.

The results show that the proposed controller seems to be a prospective solution: It is based on a simple 3rd order polynomial approach of the quasi-stationary states and

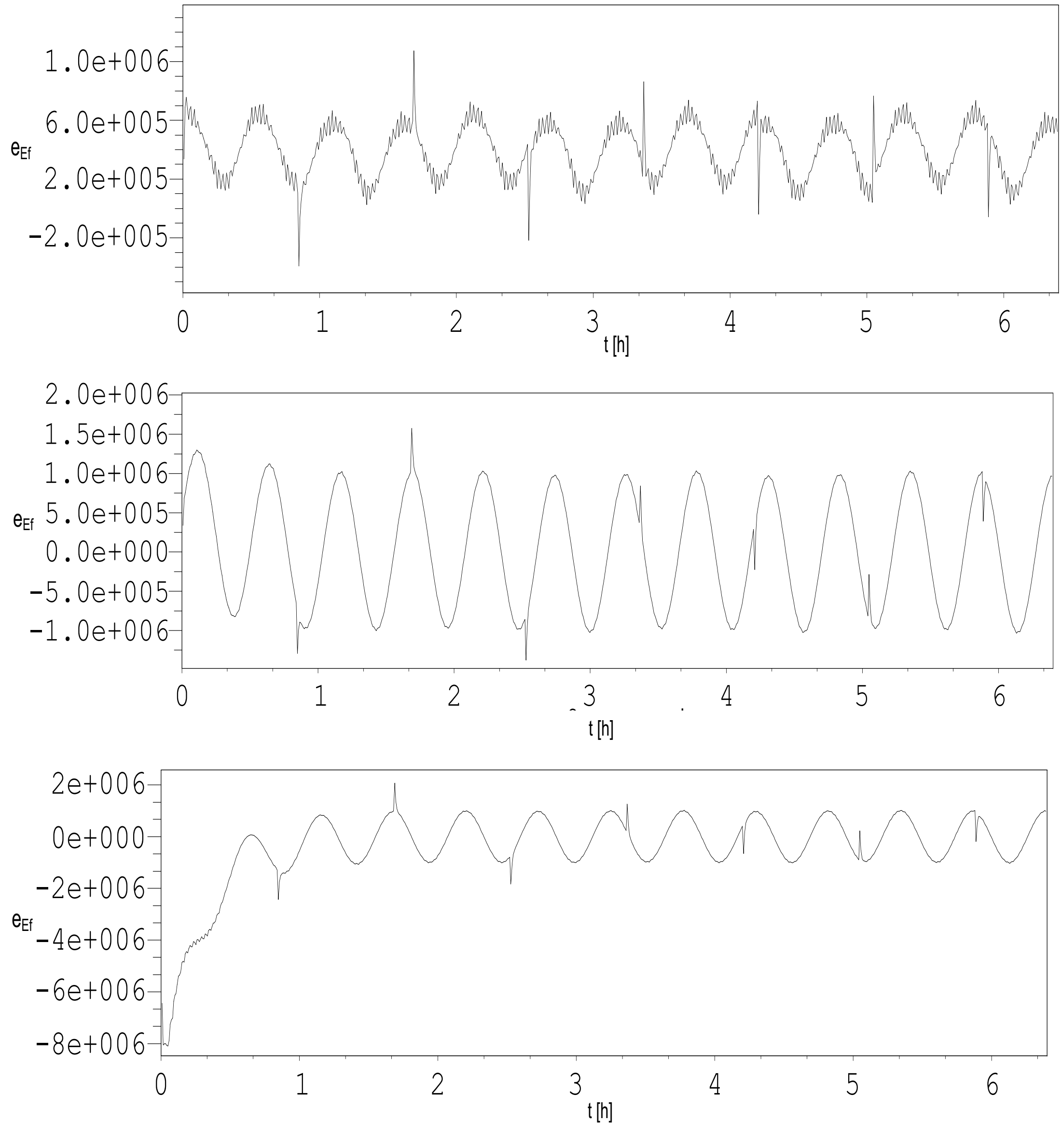


Figure 11.7: Ex1.: Tracking errors of the emission factor with exact model parameters using controller C_1 (upper); with exact model parameters using controller C_2 (middle); with approximate model parameters using controller C_2 (lower; E_f in km^2/h^3 units). When using C_1 there is a permanent error component (it is strongly shifted), otherwise there is not. The parameter approximation results in a slight increase in the error.

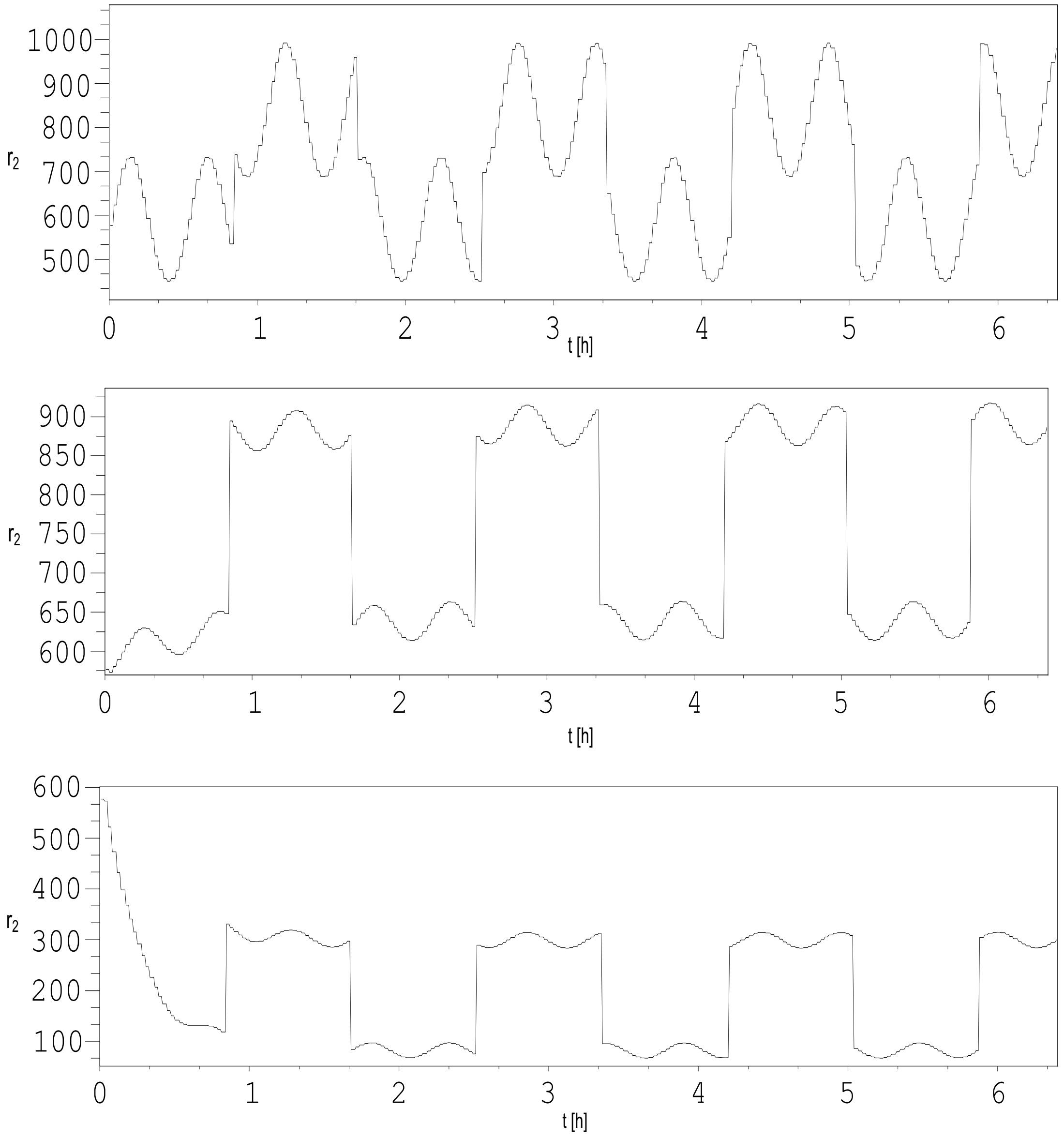


Figure 11.8: Ex1.: Illustration of the control signal (additional vehicles let into the system from the ramp in road segment 2 r_2): with exact model parameters using controller C_1 (upper); with exact model parameters using C_2 (middle); with approximate model parameters using C_2 (lower).

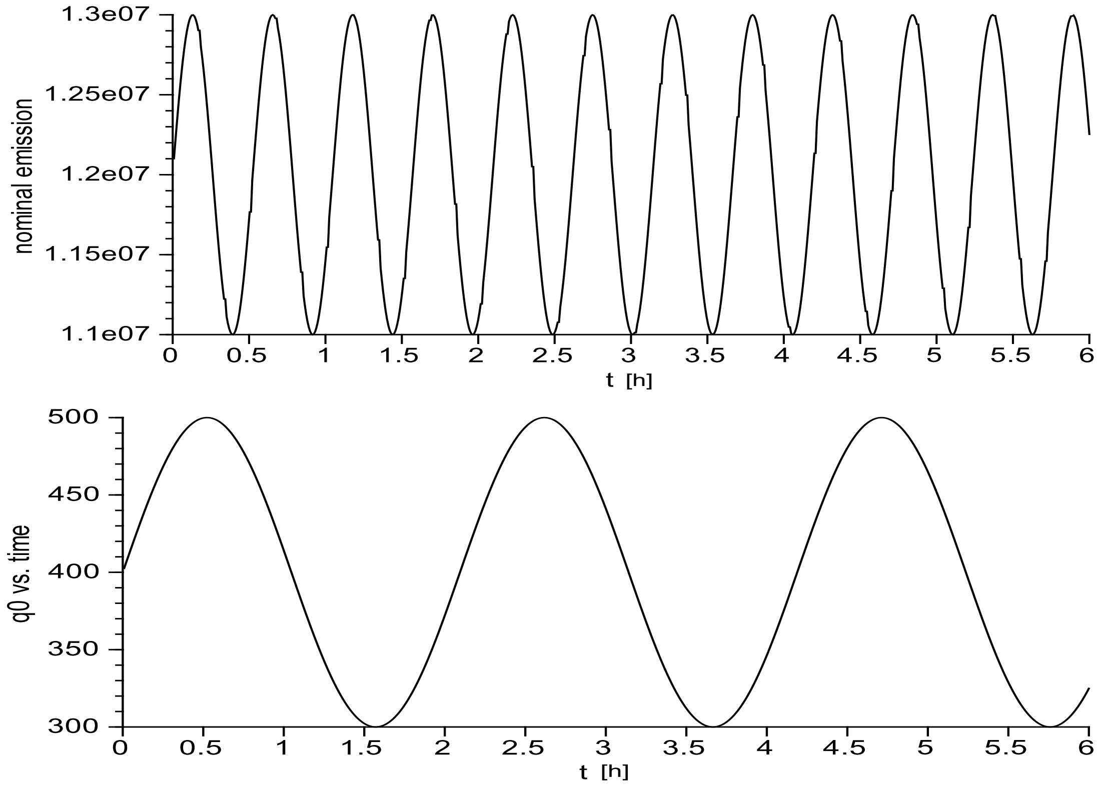


Figure 11.9: Ex2.: The nominal emission factor (upper; in km^2/h^3 units) and the variation of $\hat{q}_0$ (lower). In this example, both vary continuously.

the transformation of RFPT. It applies the ingress rate from a ramp in the preceding road segment as control signal and requires only the measuring of the traffic velocity and vehicle density in the controlled segment. The proposed method applies offline processing of the available analytical model for the determination of the stationary state. The control process can be solved by common, commercially available softwares that do not require more computational capacity than that of a common laptop or a PC. So the real-time computations need very little computational capacity if the model detailed in Section 3.7 is used. Since the suggested method needs continuous observations, it cannot ensure “asymptotic stability” (any correction is possible only after the observation). It can guarantee only stability (in Lyapunov sense).

The results considered to be new have been published in journal paper [J1] and conference papers [C16, C17, U1].

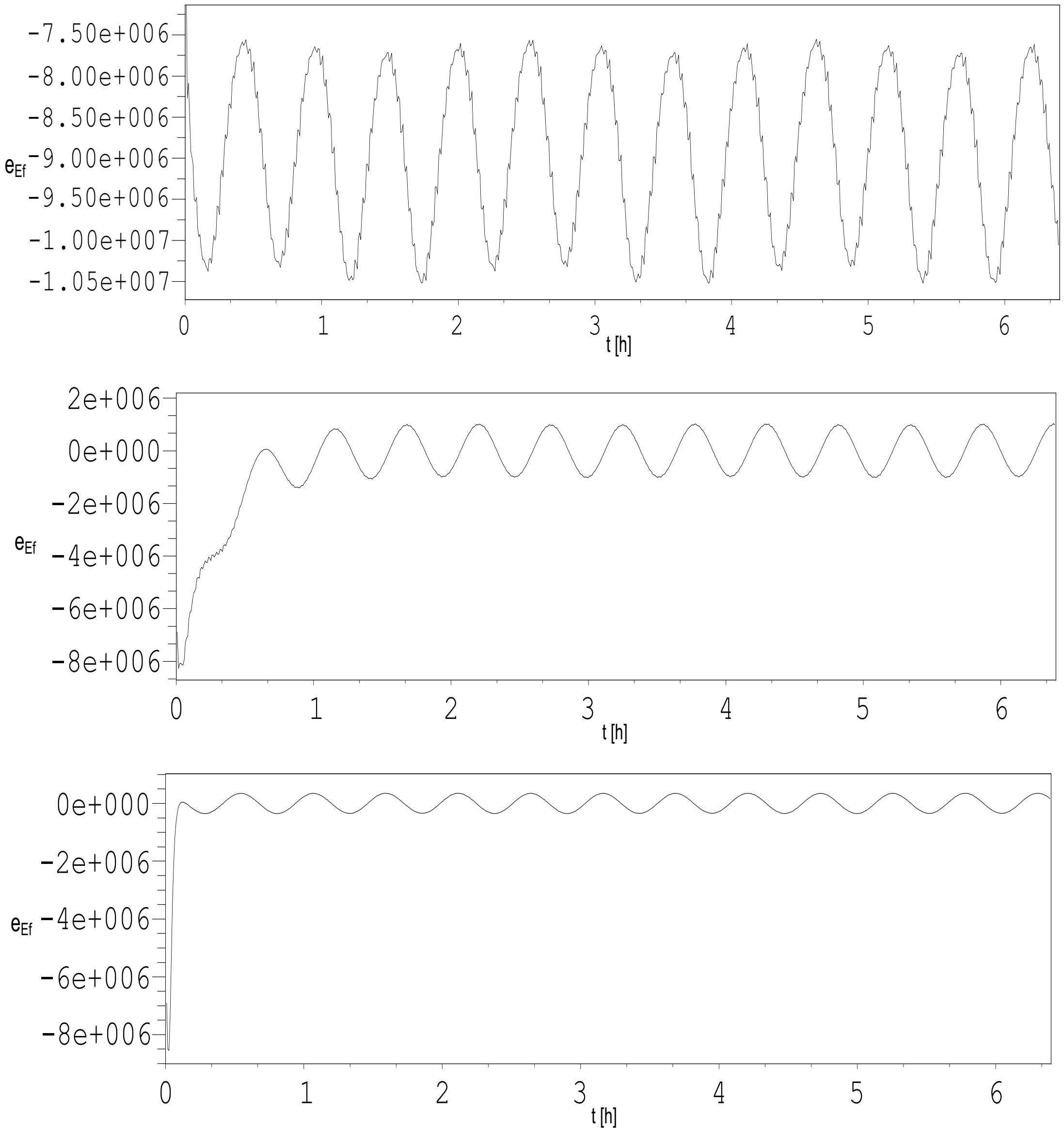


Figure 11.10: Ex2.:Tracking errors of the emission factor with exact model parameters using controller C_1 (upper); with exact model parameters using controller C_2 (middle); with approximate model parameters using controller C_2 (lower; E_f in km^2/h^3 units). When using C_1 the error is shifted, otherwise it is not. The low sampling time results in error reduction.

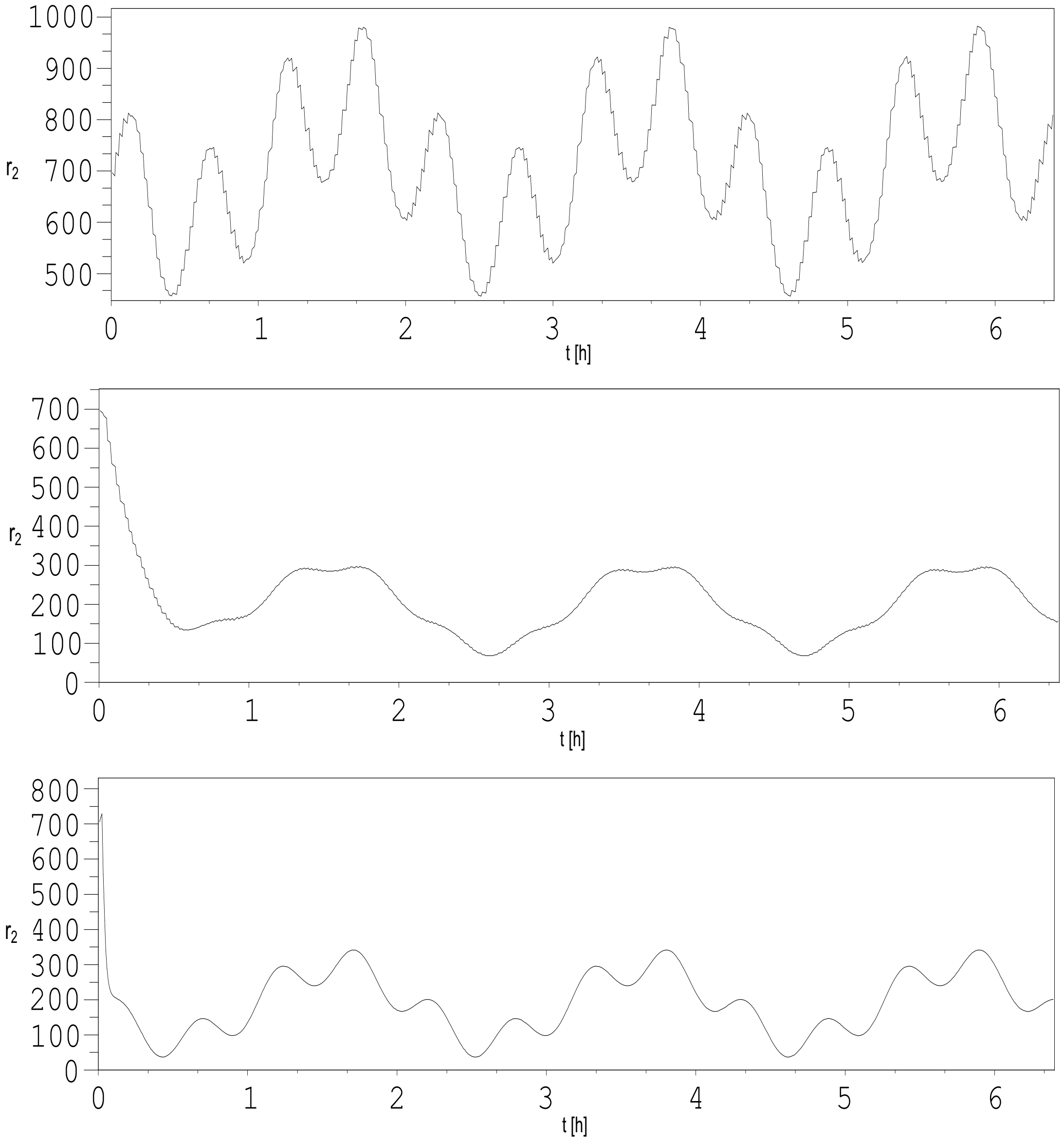


Figure 11.11: Ex2.: Illustration of the control signal (additional vehicles let into the system from the ramp in road segment 2 r_2): with exact model parameters using controller C_1 (upper); with exact model parameters using C_2 (middle); with approximate model parameters using C_2 (lower).

12

Anti-lock braking system

In the previous chapter, a real life problem has been attached. An approximate hydrodynamic model has been adapted, analyzed, and used for the RFPT-based control of a certain freeway traffic. Following the focusing on real-life problems, in this chapter, as a preliminary investigation, a new type of anti-lock braking system is introduced which uses only the measurable parameters of a car. The greatest advantage of the proposed system is that it is far simpler than any of the ABS systems found in the literature. The proposed approach uses the approximation of a vehicle model, so the new model and controller might be considered as a preparation for a new application of RFPT in real life. The results show that despite the rough approximation, the proposed structure works well and generates similar results to real cars of today.

12.1 Introduction

Friction is a phenomenon which is present in everyday life. Without it, many types of everyday motions would not be possible (think e.g. on rolling, stepping, ...) thus, it is a basic and determining force that has helped the formation of our culture. In real life, it is present everywhere so it has to be taken in account in every task that involves moving. Its advantages have to used, while its disadvantages have to be avoided.

As a disadvantage, in control tasks, friction is a strongly nonlinear phenomenon which generates only undesirable external disturbances that has to be compensated by the controllers. E.g. in the low velocity regime of mechanical devices it causes the so-called stick–slip phenomenon resulting in unwanted oscillations in the system. For the compensation various methods have been proposed starting from the nineties to the recent past (see e.g. [118, 119]).

As an advantage, in case of driving, friction is a primary factor to determine the limits of the vehicle's acceleration and deceleration. The possible maximum acceleration and deceleration of the vehicle body is determined by a parameter called friction coefficient. The friction coefficient describes the contact of the road and the wheels, therefore its behavior is extensively studied for various tire types under various environmental conditions. From around 1990 scientists realize the importance of another variable that determines the maximal friction: it is called wheel slip. Wheel slip is defined as the ratio of the wheel's rotational speed and that of an equivalent free-rolling wheel. It is possibly the most important factor on which the available maximal friction coefficient depends [67, 120]. And yet, from physical point of view, the wheel slip may suffer from criticism since it does not convey information on the relative velocity of the tire's contact point and the road surface. Though, it has great fundamental significance. The problem is widely investigated and several results are summarized in [68, 69, 70, 72] and later in e.g. [71, 121, 122, 123, 124].

Besides modeling friction, considerable efforts are exerted for developing efficient controllers for braking systems. Here only some of the characteristic approaches are enumerated. In [125] the effect of the active suspension system is investigated. In [126] an eddy current-based brake by wire system is designed with especial emphasis on the empirical data that seems to be strongly dependent on the testing environment.

The sliding mode controllers offer widely studied possibilities for braking purposes. In the early approaches (see e.g. [127] and [128]), the optimal slip value is assumed to be a priori known. In [129] an optimum search technique for wheel slip is applied. Possibilities for the reduction of chattering are studied e.g. in [130, 131]. In [132] typical friction coefficient – slip curves are obtained based on numerous experiments.

Soft-computing approaches are also used in braking: In [133, 134] fuzzy sets, in [135] genetic neural fuzzy control, in [136] self learning techniques and in [137] learning neuro–fuzzy techniques are applied. Grey system modeling is an alternative successful possibility for developing some abilities for automatic prediction applied in braking control [138].

After the extensive investigations of the friction models, the reader can be convinced that simple, efficient, and reliable braking control cannot be constructed based on modeling and / or identifying the parameters of the actual friction conditions. However, the previously developed friction models well describe certain qualitative properties of tire friction. So it seems to be more viable way to construct a controller that works by simply observing the qualitative properties of the motion. In the following, a simple vehicle model and a control strategy are proposed based on the qualitative properties of the friction explained in Section 3.8.

12.2 The vehicle model and the suggested control approach

For constructing a simple vehicle model, it is assumed that the car has four wheels in more or less symmetric positions. Then the proposed model can be described by the following Euler-Lagrange equations:

$$\begin{aligned}\dot{\omega} &= \left[rF_z\mu(\omega,v) - B_w\omega - T_b\right]/J\\ \dot{v} &= \left[-4F_z\mu(\omega,v) - B_v v\right]/m\\ \dot{T_b} &= \left[-T_b + K_bP_b\right]/\tau_{abs}\end{aligned} \tag{12.1}$$

where F_z denotes the vertical contact force (in this case it is assumed to be constant but if the cushion/swinging of the chassis in the vertical direction is taken into account it may depend on time to some extent); J denotes the constant inertial momentum of the wheel/wheel shaft/motor system; m is the mass of the car; B_v and B_w are viscous friction coefficients; P_b denotes the pressure in the braking system (in this case in arbitrary units); K_b is a gain constant; T_b marks the braking torque; and τ_{abs} denotes the time constant of the hydraulic braking system. The other quantities are defined as in Section 3.8 (ω: rotational velocity of the wheel axis; r: radius of the wheel; μ: friction coefficient; v: velocity of the vehicle). It is reasonable to assume that ω can be measured directly, while $\dot{v}$ is also measurable by cheap acceleration sensors (due to skidding ω and v are independent quantities). Altogether it means that every variable in these equations can be measured by sensors. This makes the model very simple.

For controller design, it is assumed that m, B_v, B_w, and τ_{abs} are approximated by $\hat{m}$, $\hat{B}_v$, $\hat{B}_w$, and $\hat{\tau}_{abs}$, respectively. This is a reasonable supposition since the vehicle's full mass depends on the actual payload while viscous friction may depend on environmental conditions as temperature, etc. The gravitational acceleration g is also approximated as $\hat{g}$. It is reasonable to assume that the other quantities (J, r, K_b) can be precisely known. For $\mu(\omega, v)$ no model is used due to its unreliable nature. Instead of that an estimation for the quantity $\vartheta_{abs} := F_z\mu$ is used as

$$\vartheta^{Est} = -(\hat{m}\dot{v} + \hat{B}_v v)/4 \tag{12.2}$$

where $\dot{\vartheta}^{Est}$ can be monitored. For slowly varying F_z this roughly corresponds to $F_z\dot{\mu} = F_z\frac{\partial\mu}{\partial\omega}\dot{\omega} + F_z\frac{\partial\mu}{\partial v}\dot{v}$. If $\dot{\vartheta}^{Est} > 0$ then it is reasonable to further decrease $\dot{\omega}$ assuming that the prescribed deceleration of the car body have not yet been achieved. For this purpose a maximal deceleration for ω can be prescribed as $\ddot{\omega}^{Des} = -D_\omega^{Max}$. If the prescribed deceleration for the car body D_v^{Max} is achieved or ϑ^{Est} cannot be increased, then ω should be stabilized at its present value by the control rule

$$\ddot{\omega}^{Des} = -\Lambda\dot{\omega} \tag{12.3}$$

where Λ is a positive constant. In the possession of the appropriate $\ddot{\omega}^{Des}$ and the estimated parameters, the approximate $\dot{T}_b^{Est}$ can be calculated with the help of the first equation in (12.1):

$$\dot{T}_b^{Est} = (-J\ddot{\omega}^{Des} + r\dot{\vartheta}^{Est} - \hat{B}_w\dot{\omega}). \tag{12.4}$$

$\dot{T}_b$ can be substituted to the third equation of (12.1) by $\dot{T}_b^{Est}$. With the approximate parameters the approximate braking pressure is achievable. Since it cannot be negative, the proper expression is

$$P_b^{Est} = max[0, (\hat{\tau}_{abs}\dot{T}_b^{Est} + T_b^{Est})/K_b] \tag{12.5}$$

The concrete braking torque T_b can be obtained by numerical integration of the third equation of (12.1) and using the approximate value $\hat{\tau}_{abs}$.

The block scheme of the above described system is shown in Fig. 12.1.

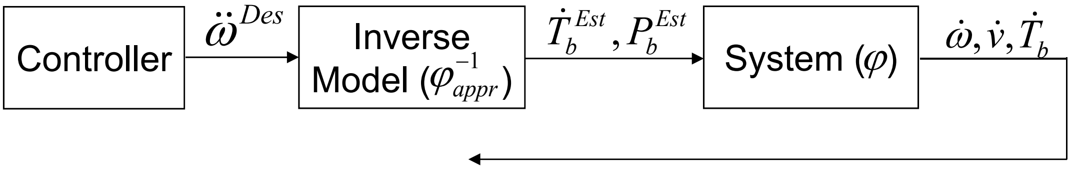

Controller
$\ddot{\omega}^{Des}$
Inverse Model (φ_{appr}^{-1})
$\dot{T}_b^{Est}, P_b^{Est}$
System (φ)
$\dot{\omega}, \dot{v}, \dot{T}_b$

Figure 12.1: The block scheme of the proposed anti-lock braking system.

12.3 Simulation results

In this section, some illustrative simulation results are presented without limiting the generality of the proposed method. The simulations of the anti-lock braking are made in Scilab-SCICOS environment. In the examples, the numerical parameters are, as follows: $B_v = 0.5\,Ns/m$, $B_w = 0.05\,Nms/rad$, $m = 1600\,kg$, $r = 0.25\,m$, $J = 500\,kgm^2$, $\tau_{abs} = 0.001\,s$, $g = 9.81\,m/s^2$, $F_z = mg/4$, $K_b = 0.1$, $\hat{B}_v = 1\,Ns/m$, $\hat{B}_w = 0.1\,Nms/rad$, $\hat{m} = 2000\,kg$, $\hat{t}au = 0.0015\,s$, $\hat{g}_{=}10\,m/s^2$, and $\hat{F}z = \hat{m}\hat{g}/4$. The controller's parameter settings are: $D_v^{Max} = 3\,g$, $D_\omega^{Max} = 100\,rad/s^3$, and $\Lambda = 500/s$. In the numerical simulations the exact value of μ is calculated from (3.27) with the numerical approximation of $1/v$ as $v/(a^2 + v^2)$ with $a = 10^{-3}\ m/s$ to avoid numerical singularities at $v = 0$. The initial velocity of the (free rolling) car is $v_{ini} = 52\,m/s$.

In the first example, it is assumed that braking is initiated on dry asphalt that later becomes wet, snowy, wet, and dry again. The variation of v, ω, and μ during braking is illustrated in Fig. 12.2. The changing road conditions can be well observed: e.g. on snowy asphalt the velocity is almost constant and the friction coefficient is very low. The figure reveals that in the beginning, considerable braking action is possible even on the wet asphalt session. The stagnation in v after $12\,s$ may be related to the low relative velocity. Figures 12.3 reveals the pressure on the braking system. It can be seen that the braking process is very similar to that of real cars' real ABS. Figure 12.4 shows that the relative velocity is kept at a relatively high value during the whole braking session. Finally, in Fig. 12.5 the braking distance can be seen.

In the second example, the simulation is made on dry asphalt. Compared to the changing road conditions, Fig. 12.6 reveals that the friction coefficient is kept at relatively high value with even more braking (Fig. 12.7) and even more relative velocity (Fig. 12.8). The braking route becomes considerably shorter, too (see Fig. 12.9). The braking distance is similar to a Lotus Elise S2's non official braking route which does not have ABS (see [66]).

In the case of evenly dry asphalt the variation of the estimated $\dot{\vartheta}^{Est}$ is not so hectic so Fig. 12.10 provides good information on the operation of the controller.

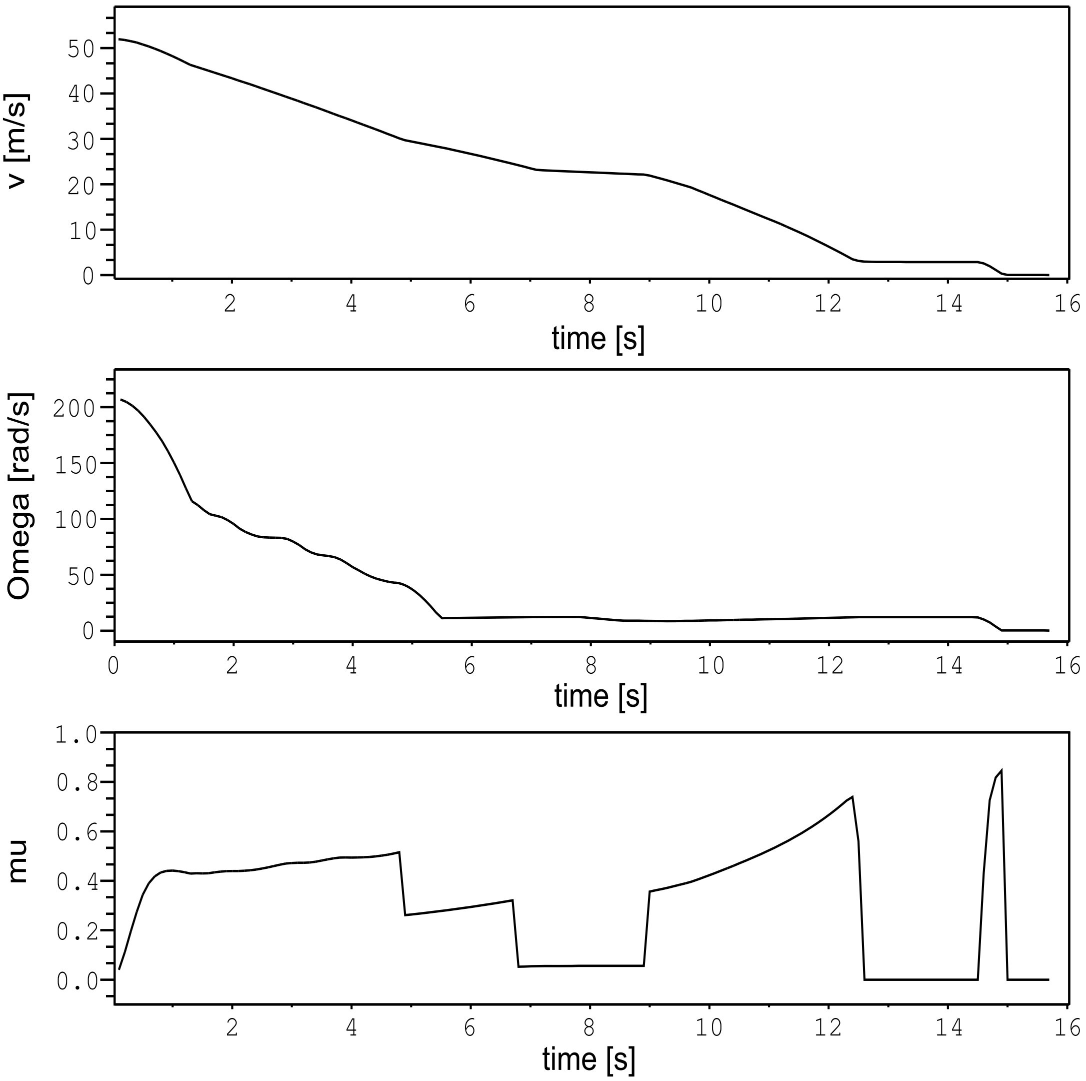


Figure 12.2: The variation of v, ω, and μ during braking in case of varying road conditions: $0-5\,s$: dry, $5-7\,s$: wet, $7-9\,s$: snowy, $9-11\,s$: wet, and $11-16\,s$: dry. The changing road conditions can be well observed, e.g. on snowy asphalt the velocity is almost constant and the friction coefficient is very low.

12.4 Summary

In this chapter, preliminary investigations are made on a possible anti-lock braking system that does not wish to use or identify any sophisticated friction model. The reason for this approach is the fact that these models are strongly nonlinear, difficult to identify, and their parameters can change suddenly with varying road conditions.

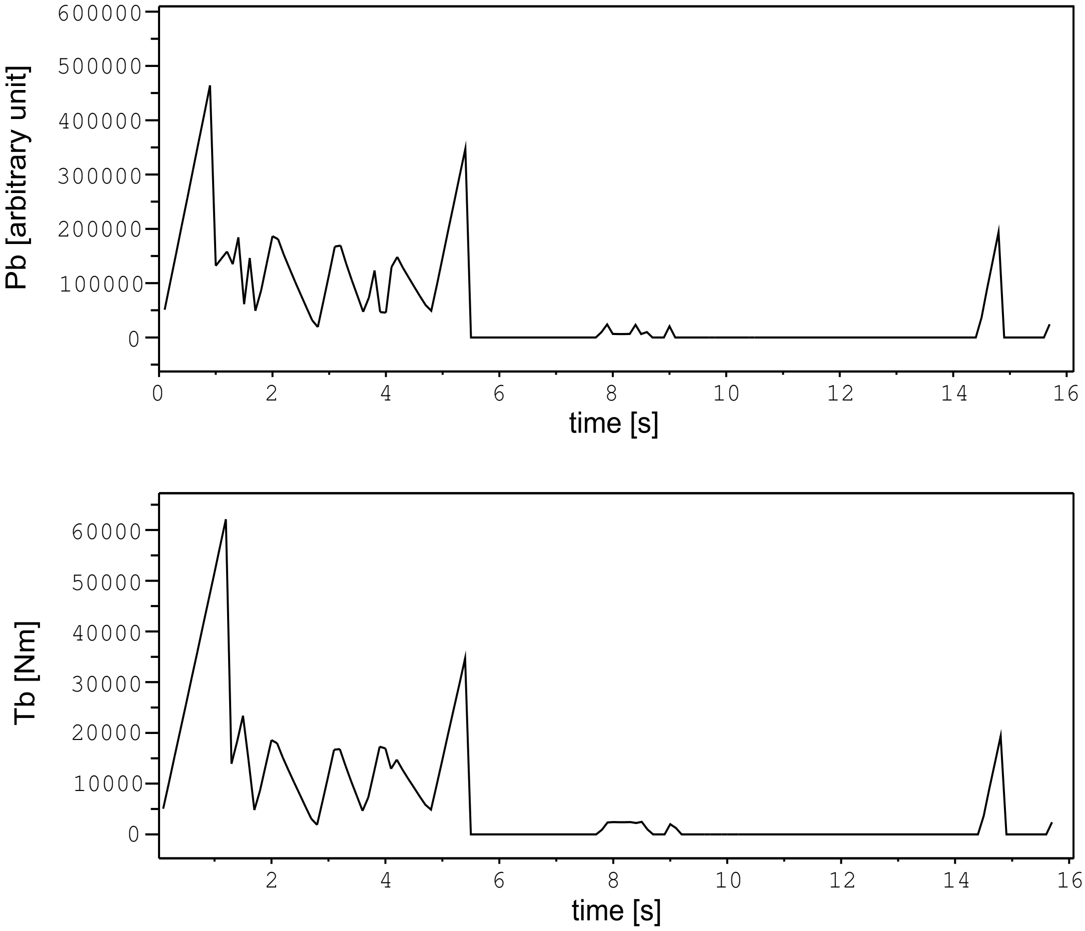


Figure 12.3: The variation of the braking pressure P_b and the braking torque T_b in case of varying road conditions. The figures reveal similar braking process to that of a real ABS system.

Instead of a friction model, a simple vehicle model is suggested together with a simple control rule. The results show that though the calculations based on the proposed model contain a lot of approximations, the controller indicates the proper control actions so that the system produces similar results to a real car used today.

The result considered to be new has been published in conference paper [C2]. Based on the proposed model the suggested approach has been extended with a new tire model and an RFPT-based controller in [139].

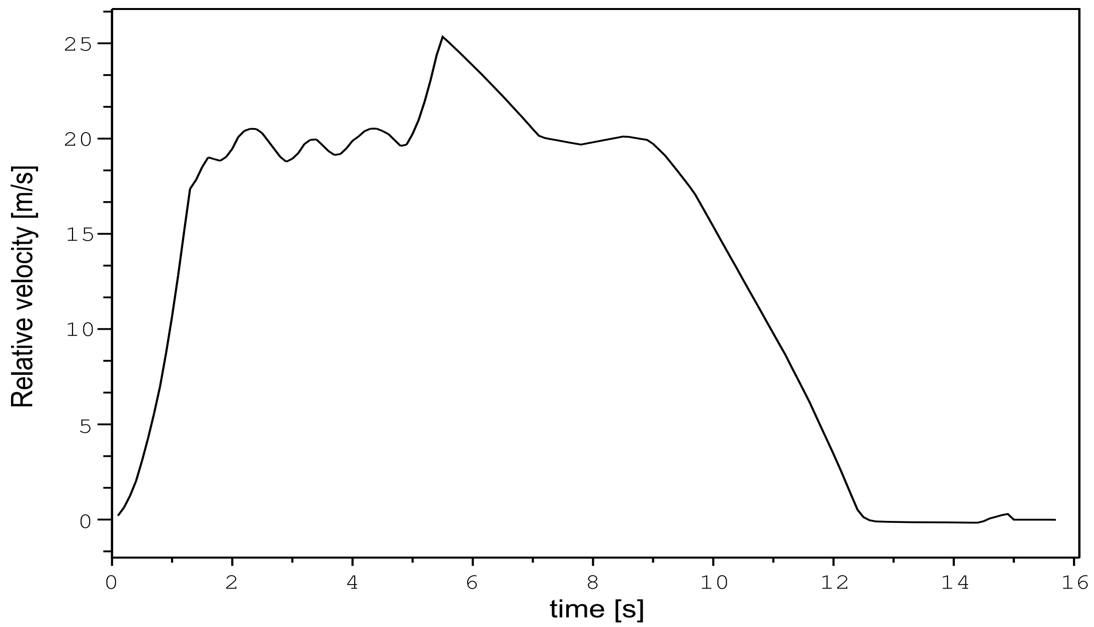


Figure 12.4: The variation of the relative velocity $v - r\omega$ in case of varying road conditions. It is kept at a relatively high value during the whole braking session.

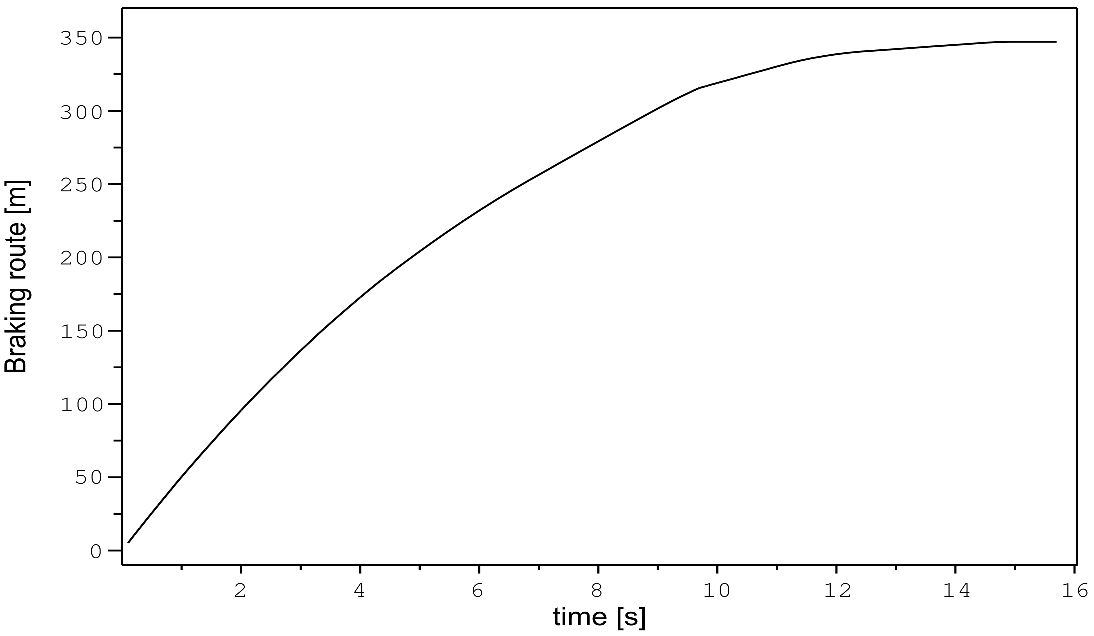


Figure 12.5: The braking distance in case of varying road conditions.

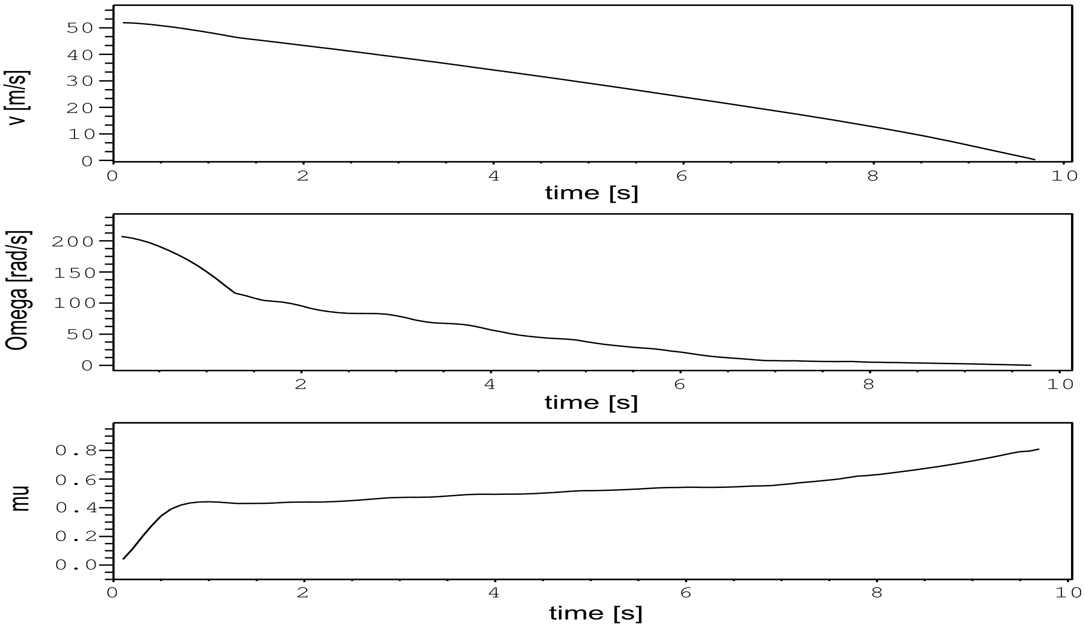


Figure 12.6: The variation of v, ω and μ during braking on dry asphalt. The velocity decrease smoothly and the friction coefficient is kept at high.

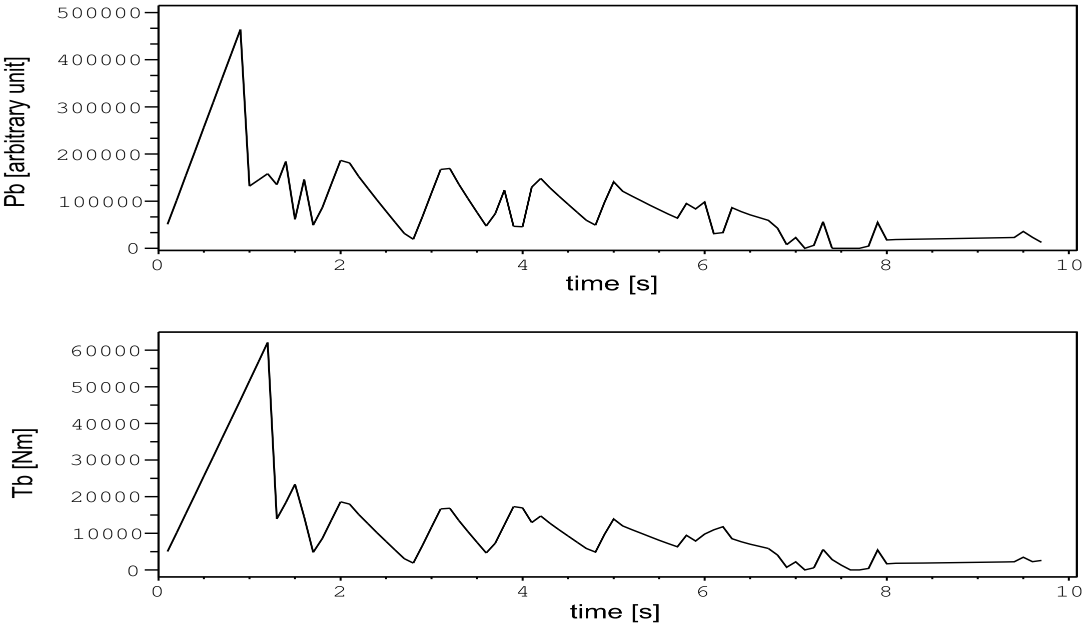


Figure 12.7: The variation of the braking pressure P_b and the braking torque T_b during braking on dry asphalt. Relatively high values can be observed during the whole session. The figures reveal similar braking process to that of a real ABS system.

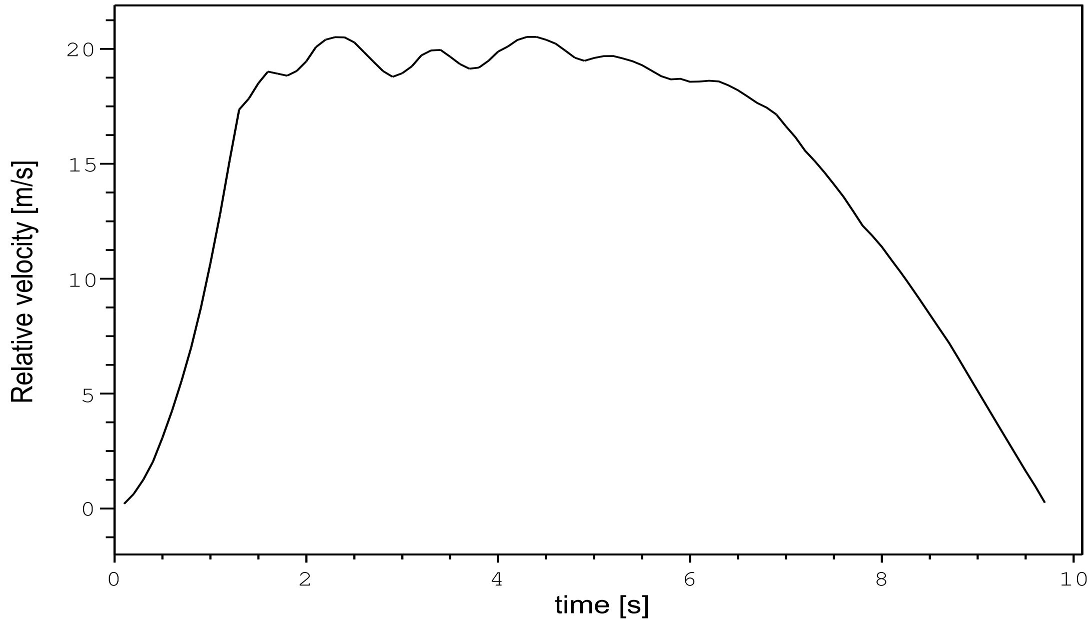


Figure 12.8: The variation of the relative velocity $v - r\omega$ during braking on dry asphalt. The relative velocity is kept high during the whole session.

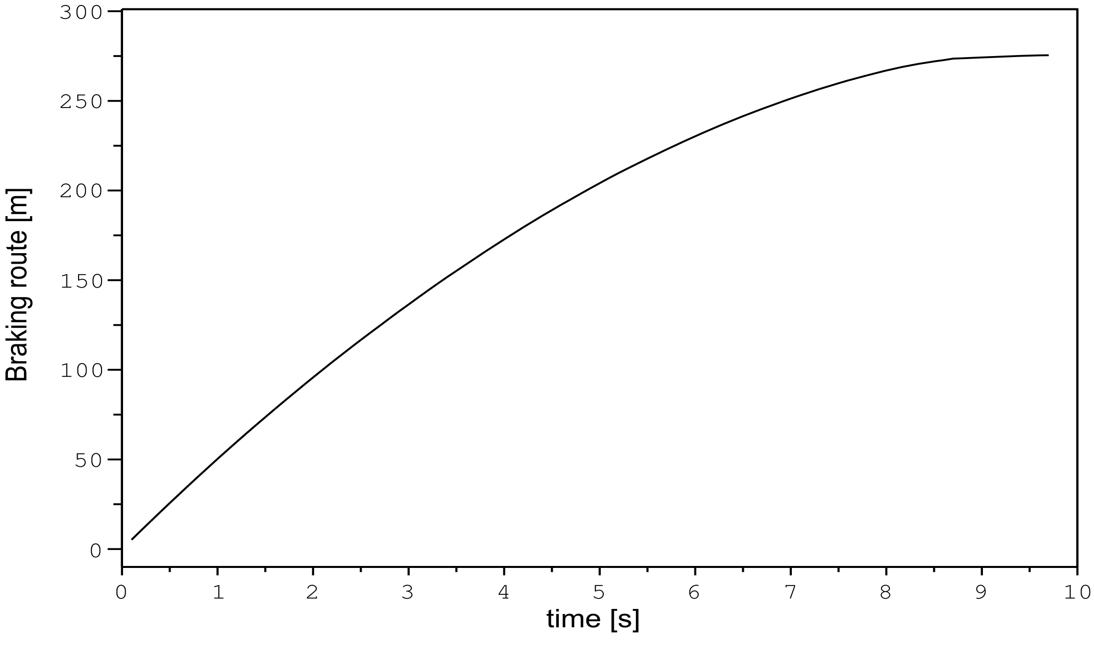


Figure 12.9: The braking distance on dry asphalt vs. time (in s units). The braking route is decreased significantly compared to the changing road conditions (see Fig. 12.5). The braking distance is similar to a Lotus Elise S2's non official braking route (see [66]).

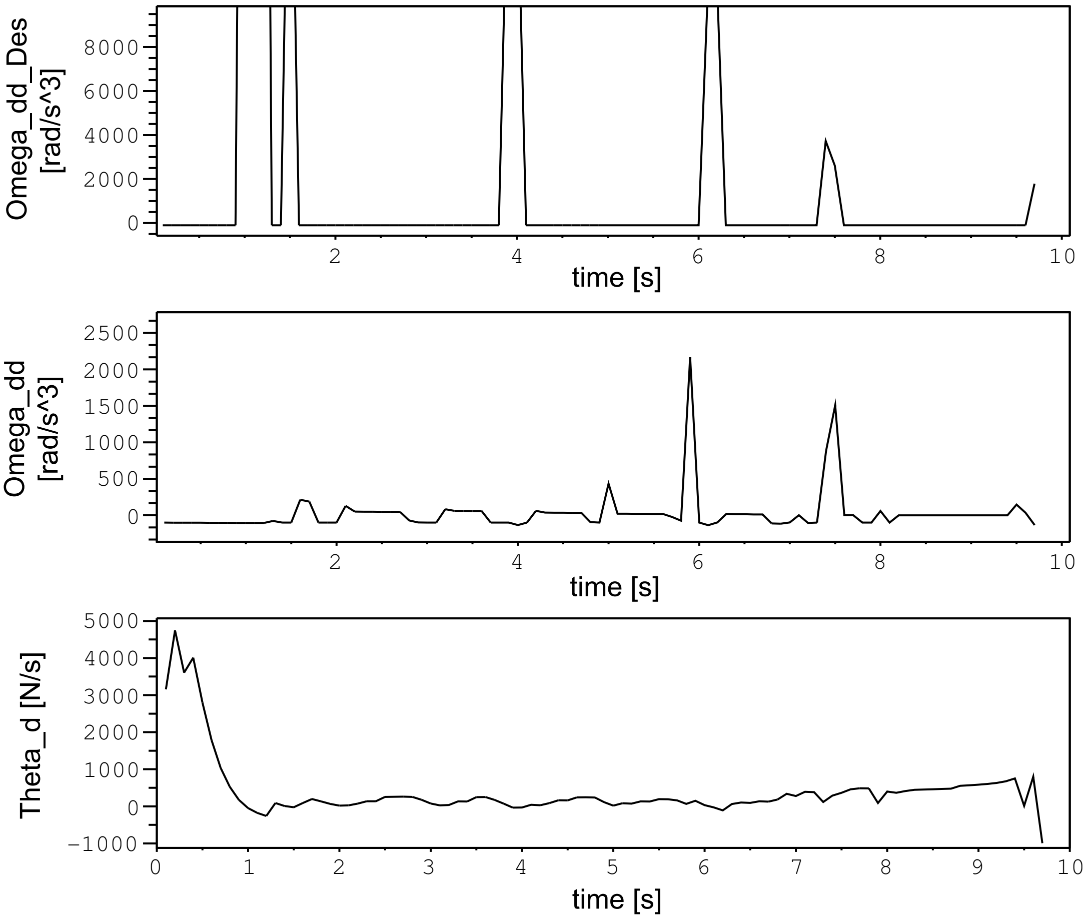


Figure 12.10: The *desired* and the *realized* value of $\ddot{\omega}$ and the the estimated $\dot{\vartheta}^{Est}$ value during braking on dry asphalt.

13

Conclusions

After detailing the results, in this chapter, the final conclusions are summarized.

13.1 The most important statements of the thesis

Nowadays, control, especially automated control of systems with uncertainties is essential in everyday life. The uncertainties can be classified in many different ways, e.g. the system can be known, partially known, or unknown; the data/information can be exact, uncertain, inaccurate, or lack of data, etc.

In this thesis, that classification is followed which considers the efficiency of the applied controllers. From this point of view, three important groups can be divided: 1. when the system contains unknown parameters 2. when the system has unknown dynamics 3. when the the system's state cannot be measured [4]. Without attempting to be comprehensive, there are some important types of controllers that fit the different situations, e.g. sliding mode controllers, fuzzy logic controllers, neural network controllers, and robust controllers, etc.

When the controlled system is just partly known robust controllers bring the most benefit so they are popular, applied, and improved in our days, too. One of the recent robust control strategies is the method called Robust Fixed Point Transformations (RFPT). It can be used both in traditional feedback control systems and Model Reference Adaptive Control systems and also for single input – single output systems and multiple input – multiple output systems. Its aim is to make controllers robust when an approximate model is used in the control process to estimate the behavior of the system.

This dissertation focuses on improving RFPT, because

1. It guarantees only local stability of the controller according to Lyapunov's stability theorem;
2. Up till now it has been applied to ameliorate only traditional controllers;
3. It can a be a powerful method to analyze and control systems with modeling difficulties (either because the analytical model is not known or is too complex to be used).

First, in Chapter 5, a new application area is introduced for Robust Fixed Point Transformations: the chaos synchronization. Several chaotic oscillators are analyzed, modeled, and controlled with RFPT to show that the improved controllers achieve more precise trajectory tracking.

Based on the main focus and the three possible positions of the RFPT in the control loop, a new structure for RFPT is introduced. The structure is based on the idea of integrating a further controller into the system. The second controller gains an extra tracking error reduction compared to the original structures without increasing the computational burden significantly. This results in a more precise trajectory tracking.

In the third part (Chapters 7 and 8), two methods are proposed that make the RFPT-based controllers stable. First, a fuzzy-like parameter tuning, then a VS-type stabilization method is suggested to make sure that if the RFPT-based controllers loose their stability then it is gained back within a very short time. The two methods make the RFPT-based controllers stable and because of this, more trustworthy.

In the next step, RFPT's joint applicability is verified with two of the increasingly prevalent soft-computing-based controllers: first with a fuzzy logic controller then with a neural network controller. The results show that the robustness of these controllers can be increased with RFPT and because of that the RFPT-based soft-computing-based controllers give more accurate results than the original ones. The two approaches prove that RFPT can advantageously be used for solving recent problems with current techniques and gives an opportunity to improve two more and more popular controllers.

Finally, a further aspect, the applicability of RFPT in complex real-life systems is attached: first, a hydrodynamic model of freeway traffic is analyzed from the viewpoint of stability which has only two variables (traffic density and velocity) and controlled with RFPT-based controllers in quasi-stationary cases, then a simple model for an anti-lock braking system is designed and controlled. Besides that the constructed models have only partial information about their environment, they show similar behavior to real life. The results show that if a system cannot be modeled accurately because of the lack of resources or knowledge, approximate models also can be used successfully. The approximate model has the advantage that it is less complex and thus, simple controllers can be designed for them. The inaccuracy of the controllers then can be ameliorated by combining them with adaptive techniques, like RFPT. By this, the models can be forced to behave more similar to the real system than without the suggested adaptive improvements.

13.2 The new scientific results of the thesis

I. Thesis Robust Fixed Point Transformations in chaos synchronization.

I proposed a new application area for Robust Fixed Point Transformations: the chaos synchronization. Based on the literature I studied the most important chaotic attractors (FitzHugh-Nagumo, Matsumoto-Chua, Duffing, and Van der Pol oscillators) and I designed approximate models and controllers for them. I showed that if the parameters of a chaotic system are not known exactly or external disturbances affect the system, then RFPT is an effective method to get more accurate trajectory tracking [J2, C1, C4, C5, C6, C7, C8, C19].

II. Thesis Mathematical development for Robust Fixed Point Transformation.

I analyzed the mathematical background of Robust Fixed Point Transformations. I proposed a new structure for RFPT in which two controllers are integrated into the system. I showed that the new structure gains an additional tracking error reduction compared to the original RFPT methods [J2].

III. Thesis group Stability of the Robust Fixed Point Transformations-based controllers.

3.1 I considered the stability of the Robust Fixed Point Transformations-based controllers and I introduced an innovative fuzzy-like parameter tuning method for RFPT. I showed that more stable results of RFPT can be gained if the fuzzy-like parameter tuning is applied in the control process [C12, C13, L1, C18].

3.2 I reconsidered the stability of the Robust Fixed Point Transformations-based controllers and I proposed a new VS-type stabilization algorithm for RFPT. I showed that when the RFPT-based controller falls out from the local interval of convergence it becomes unstable and generates the so-called chattering effect. I showed that the proposed algorithm can reduce the order of fluctuation and can stop chattering within very short time. As a consequence the stability of the RFPT-based controllers can be gained [C19].

IV. Thesis group Joint applicability of Robust Fixed Point Transformations.

4.1 I suggested the combination of fuzzy logic controllers (FLC) and RFPT and I designed an RFPT-based fuzzy logic controller. I compared the performance of the FLC and that of the RFPT-based FLC and I verified via simulations that the robustness of the original FLC can be increased with the application of RFPT and by this the error produced by the original fuzzy logic controller can be reduced significantly [C20].

4.2 I suggested the combination of neural network controllers (NNC) and RFPT and I designed an RFPT-based neural network controller. I compared the performance of the NNC and that of the RFPT-based NNC and I verified via simulations that the robustness of the original NNC can be increased with the application of RFPT and by this the error generated by the original neural network controller can be reduced significantly [J4].

V. Thesis group The applicability of Robust Fixed Point Transformations in complex, partially known systems.

5.1 I investigated the applicability of Robust Fixed Point Transformations in complex, partially known systems. For this, I studied a hydrodynamic model of freeway traffic. I determined the stationary solutions of the model, and analyzed their stability. I introduced a new attribute which is strongly related to the emission rate of exhaust fumes of the freeway traffic (and based on the new attribute the control strategy can be more simple). I designed an RFPT-based controller to control the emission rate of exhaust fumes for the quasi-stationary solutions based on the new attribute connected to the emission rate. I showed that the RFPT-based controller is able to limit the emission rate with proper control actions, while the original one (without RFPT) cannot [J1, C16, C17,U1].

5.2 I investigated a further new area where the Robust Fixed Point Transformations method is a promising candidate to improve the control performance. I suggested a simple vehicle model to be used in anti-lock braking systems. I showed that the introduced model is simpler (both from the point of view of complexity and that of the measurability of the considered effecting coefficients) than any effective model known from the literature. I designed a controller which uses only the suggested rough approximate model and showed that good results can be gained with it [C2].

13.3 Application and future work

Recently, model-based approaches have proved to be very advantageous in control tasks. Although, many problems arise from the high complexity of the tasks and also from the inaccuracy, uncertainty, and vagueness of our knowledge about the systems to be controlled. In such situations, classical control methods may fail to work.

Robust controllers are one of the best solutions to handle uncertain systems, among which the Robust Fixed Point Transformations-based controllers are outstanding tools. They can advantageously be used to restructure and improve existing controllers that use approximate models for the control process. In this thesis, different application areas, e.g. chaos synchronization and traffic control, are suggested (and examples are shown) where RFPT can be a good candidate. Similar considerations can be made in other fields of engineering, like industry, manufacturing, and machines (etc.), where controllers are widely used and the uncertainties of the system approximations have to be handled.

The topics concerned by this dissertation leave several open questions that need further research:

1. The VS-type stabilization method of RFPT deals with the decrease of the possible A_i parameters to regain the stability of the RFPT-based controllers. New algorithms might be more effective which, in certain cases, can also increases these parameters to achieve more accurate results.

2. Since the exact relationship between parameters A and K is not determined yet (it strongly depends on the control task), further parameter tuning methods, e.g. neural network tuning, could be used to calculate their appropriate values.

3. The dissertation investigates only two possibilities of improving soft-computing-based controllers. Other approaches could be analyzed, too.

4. For the emission control of exhaust fumes of freeway traffic, fitted polynomial packages could be prepared for several reasonable ρ_0, v_0, and $v_4 = v_5$ combinations depending on some "typical" traffic situations, since the boundary conditions significantly influence the behavior of the solutions.

5. For the emission control problem, higher number of road segments may lead to better approximation of the continuum model, e.g. receding horizon controllers (also known as model predictive controllers, see [140]) could be used.

6. The anti-lock braking system, shown in this thesis, does not allow the direct application of adaptive techniques. Thus, a possible development could be restructuring and applying improver techniques, like RFPT, in this approach. (In this direction, the first step has already been taken. In [139], the authors, based on my approach, extended their tire model with RFPT).

13.4 Publications of the author strongly related to the scientific results

13.4.1 Journal papers (international refereed periodicals)

J1. J. K. Tar, L. Nádai, I. J. Rudas, T. A. Várkonyi, "Adaptive Emission Control of Freeway Traffic Using Quasi-Stationary Solutions of an Approximate Hydrodynamic Model," *Journal of Applied Nonlinear Dynamics (JAND)*, 1(1), pp. 29–50, 2012, ISSN: 2164-6457.

J2. T. A. Várkonyi, J. K. Tar, I. J. Rudas, "Robust Fixed Point Transformations-based Control of Chaotic Systems," *Computing and Informatics (CAI)*, 32(6), 2013, ISSN: 1335-9150, Indexed by SCOPUS and SCI, IF=0.239, in print.

J3. T. A. Várkonyi, J. K. Tar, Imre J. Rudas, "Improved Stabilization for Robust Fixed Point Transformations-based Controllers," *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, 17(3), pp. 418–424, 2013, ISSN: 1343-0130, Indexed by Scopus and EI, in print.

J4. T. A. Várkonyi, J. K. Tar, Imre J. Rudas, "Improved Neural Network Control of Inverted Pendulums," *International Journal of Advanced Intelligence Paradigms (IJAIP)*, ISSN: 1755-0386, Indexed by SCOPUS and INSPEC, accepted.

13.4.2 Journal papers (local refereed periodicals)

U1. J. K. Tar, I. J. Rudas, L. Nádai, T. A. Várkonyi, "A közúti közlekedés kvázistacionárius adaptív, iteratív szabályozása két ellentmondó kritérium szerint," (Quasi-stationary, Adaptive, and Iterative Control of Freeway Traffic by Two Inconsistent Criteria) *Közlekedéstudományi szemle*, 52(3), pp. 42–48, 2012, ISSN: 0023-4362 (In Hungarian).

13.4.3 Conference papers (international refereed conferences)

C1. T. A. Várkonyi, J. K. Tar, I. J. Rudas, "Robust Fixed Point Transformations in Chaos Synchronization," In: *Proc. of the 11th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, Budapest, Hungary, pp. 219–224, 2010, ISBN: 978-1-4244-9278-7.

C1 – C1 S. John, J. O. Pedro, L. T. Kóczy, “Adaptive Improvement of a Passive Antilock Brake Control,” In: *Proc. of IEEE AFRICON*, Livingstone, Zambia, pp. 1–6, 2011.

C2. T. A. Várkonyi, J. F. Bitó, I. J. Rudas, J. K. Tar, “Preliminary Investigations on a Higher Order Modelfree Approach in Antilock Braking,” In: *Proc. of the 9th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Smolenice, Slovakia, pp. 259–263, 2011, ISBN: 978-1-4244-7428-8.

C2 – C1 S. John, J. O. Pedro, L. T. Kóczy, “Adaptive Improvement of a Passive Antilock Brake Control,” In: *Proc. of IEEE AFRICON*, Livingstone, Zambia, pp. 1–6, 2011.

C3. J. K. Tar, I. J. Rudas, T. A. Várkonyi, K. R. Kozlowski, “Efficient and Simple Noise Filtering for Stabilization Tuning of a Novel Version of Model Reference Adaptive Controller,” In: *Proc. of the 5th International Workshop on Robot Motion and Control (RoMoCo)*, Bukowy Dworek, Poland, pp. 205–214, 2011.

C4. T. A. Várkonyi, J. K. Tar, J. F. Bitó, I. J. Rudas, “Simple Noise Reduction in the Adaptive Synchronization of Coupled Neurons by Robust Fixed Point Transformation,” In: *Proc. of the 15th IEEE International Conference on Intelligent Engineering Systems (INES)*, Poprad, Slovakia, pp. 297–302, 2011, ISBN: 978-1-4244-8955-8.

C4 – B1 F. Adamčk, R. Andoga, L. Madarász, P. Krajňák, “Elimination of Dynamic Errors of Thermocouples in Aircraft Engines Using Neural Networks,” In: *L. Madarász, Jozef Živčák (Eds.), Aspects of Computational Intelligence: Theory and Applications Topics in Intelligent Engineering and Informatics*, Springer Berlin Heidelberg, 2, pp. 185–194, 2013.

C4 – B2 R. Andoga, L. Madarász, T. Karol, L. Főző, V. Gašpar, “Intelligent Supervisory System for Small Turbojet Engines,” In: *L. Madarász, Jozef Živčák (Eds.), Aspects of Computational Intelligence: Theory and Applications Topics in Intelligent Engineering and Informatics*, Springer Berlin Heidelberg, 2, pp. 85–104, 2013.

C4 – B3 T. Lazar, L. Madarász, V. Gašpar, “The Effectiveness of Experimental Identification of Cognitive Systems,” In: *Proc. of the 11th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, Slovakia, pp. 211–214, 2013.

C5. J. K. Tar, I. J. Rudas, J. F. Bitó, T. A. Várkonyi, “Chaos Synchronization by Model Reference Adaptive Control Using Fixed Point Transformations,” In: *Proc. of the 13th IASTED International Conference on Intelligent Systems and Control (ISC)*, Cambridge, United Kingdom, pp. 23–28, 2011, ISBN-13: 978-0-88986-889-2, ISBN-10: 0-88986-889-1.

C6. I. J. Rudas, J. K. Tar, T. A. Várkonyi, “Novel Adaptive Synchronization of Different Chaotic Chua Circuits,” In: *Proc. of the Special International Conference on Complex Systems: Synergy of Control, Computing & Communications (COSY)*, Ohrid, Macedonia, pp. 109–114, 2011, ISBN: 978-9989-2175-8-6.

C7. T. A. Várkonyi, J. K. Tar, I. J. Rudas, S. Preitl, R.-E. Precup, “A Novel Approach to Robust Fixed Point Transformations,” In: *Proc. of the 5th International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII)*, Floriana, Malta, pp. 13–18, 2011, ISBN: 978-1-4577-1859-5.

C8. T. A. Várkonyi, J. K. Tar, I. J. Rudas, “Chaos Synchronization in Duffing Systems with Robust Fixed Point Transformations,” In: *Proc. of the 7th International Symposium on Intelligent Signal Processing (WISP)*, Floriana, Malta, pp. 104–109, 2011, ISBN: 978-1-4577-1401-6.

C9. T. A. Várkonyi, J. K. Tar, I. J. Rudas, “Robust Fixed Point Transformations in Model Reference Adaptive Control for a 4 DOF Classical Mechanical System,” In: *Proc. of inter-Academia*, 2011, Sucevita, Romania.

C10. J. K. Tar, J. F. Bitó, I. J. Rudas, T. A. Várkonyi, “Decentralized Adaptive Control with Fractional Order Elimination of Obsolete Information,” In: *Proc. of the 4th International Conference on Emerging Trends in Engineering and Technology (ICETET)*, Port Louis, Mauritius, pp. 43–48, 2011, ISBN: 978-1-4577-1847-2.

C11. T. A. Várkonyi, J. K. Tar, J. F. Bitó, I. J. Rudas, “Situation-dependent Adaptive Control Polynomially Eliminating the Past Information of Fading Relevance,” In: *Proc. of the 3rd IEEE International Symposium on Logistics and Industrial Informatics (LINDI)*, Budapest, Hungary, pp. 199–204, 2011, ISBN: 978-1-4577-1840-3.

C12. J. K. Tar, L. Nádai, I. J. Rudas, T. A. Várkonyi, "RFPT-based Adaptive Control Stabilized by Fuzzy Parameter Tuning," In:*Proc. of the 9th European Workshop on Advanced Control and Diagnosis (ACD)*, Budapest, Hungary, 2011.

C12 – T1 A. A. Dineva, "Nemlineáris dinamikai Rendszerek Adaptív Szabályozása Robusztus Fixpont Transzformációs Módszerrel," (Adaptive Control of Nonlinear Dynamical Systems with Robust Fixed Point Transformations Method) MSc Thesis, Óbuda University, 2013 (In Hungarian).

C13. T. A. Várkonyi, J. K. Tar, I. J. Rudas, "Fuzzy Parameter Tuning in the Stabilization of an RFPT-based Adaptive Control for an Underactuated System," In: *Proc. of the 12th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI)*, Budapest, Hungary, pp. 63–68, 2011, ISBN: 978-1-4577-0044-6.

C14. T. A. Várkonyi, J. K. Tar, I. J. Rudas, "Robust Fixed Point Transformations-based Model Reference Adaptive Control of Inverted Pendulums," In: *Proc. of the 12th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI)*, Budapest, Hungary, pp. 591–596, 2011, ISBN: 978-1-4577-0044-6.

C15. T. A. Várkonyi, J. K. Tar, J. F. Bitó, I. J. Rudas, "RFPT-based Decentralized Adaptive Control of Partially, Roughly Modeled, Coupled Dynamic Systems," In: *Proc. of the 9th IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Slovakia, pp. 35–40, 2011, ISBN: 978-1-4577-1973-8.

C16. T. A. Várkonyi, J. K. Tar, I. J. Rudas, "Adaptive Emission Control of Freeway Traffic via Compensation of Modeling Inconsistences," In: *Proc. of the 10th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herlany, Slovakia, pp. 79–84, 2012, ISBN: 978-1-4577-0195-5.

C17. J. K. Tar, L. Horváth, I. J. Rudas, T. A. Várkonyi, "Adaptive Control of Approximately Modeled Freeway Traffic by Robust Fixed Point Transformations," In: *Proc. of the 12th WSEAS International Conference on Applied Computer Science (ACS)*, Singapore City, Singapore, pp. 81–86, 2012, ISBN: 978-1-61804-092-3.

C18. J. K. Tar, I. J. Rudas, Teréz A. Várkonyi, "Simple Practical Methodology of Designing Novel MRAC Controllers for Nonlinear Plants," In: *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Kaohsiung, Taiwan, pp. 928–933, 2012, ISBN: 978-1-4673-2575-9.

C19. T. A. Várkonyi, J. K. Tar, I. J. Rudas, I. Krómer, "VS-type Stabilization of MRAC Controllers Using Robust Fixed Point Transformations," In: *Proc. of the 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, Timisoara, Romania, pp. 389–394, 2012, ISBN: 978-1-4673-1012-3.

C19 – T1 A. A. Dineva, "Nemlineáris dinamikai Rendszerek Adaptív Szabályozása Robusztus Fixpont Transzformációs Módszerrel," (Adaptive Control of Nonlinear Dynamical Systems with Robust Fixed Point Transformations Method) MSc Thesis, Óbuda University, 2013 (In Hungarian).

C20. T. A. Várkonyi, "Fuzzyfied Robust Fixed Point Transformations," In: *Proc. of the 16th IEEE International Conference on Intelligent Engineering Systems (INES)*, Budapest, Hungary, pp. 457–462, 2012, ISBN: 978-1-4673-2694-0.

13.4.4 Conference papers (local refereed conferences)

L1. J. K. Tar, L. Nádai, I. J. Rudas, T. A. Várkonyi, "Robusztus Fixpont Transzformációkra alapozott adaptív szabályozók konvergenciájának stabilizálása súlyozott átlagokkal," (Stabilization of the Convergence of Robust Fixed Point Transformations-based Adaptive Controllers with Weighted Averages) In: *Proc. of Innováció és fenntartható felszíni közlekedés (IFFK-Konferencia)*, Budapest, Hungary, pp. 255–267, 2011, ISBN: 978-963-88875-2-8 (In Hungarian).

13.5 Further publications of the author loosely related to the scientific results

13.5.1 Book chapters (international refereed books)

B1. T. A. Várkonyi, “Advantages of Fuzzy and Anytime Signal- and Image Processing Techniques - A Case Study,” In: J. Fodor, R. Klempous, C. P. S. Araujo (Eds.) *Recent Advances in Intelligent Engineering Systems*, Berlin: Springer Berlin Heidelberg, pp. 283–301, 2012, ISBN: 978-3-642-23228-2.

13.5.2 Conference papers (international refereed conferences)

F1. A. R. Várkonyi-Kóczy, T. A. Várkonyi, “Anytime modeling: Compression and improvement of the approximation of SC based mappings,” In: *Proc. of the 4th International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII)*, Luxor, Egypt, pp. 107–112, 2009, ISBN: 978-1-4244-5380-1.

F2. T. A. Várkonyi, “Soft computing-based signal processing approaches for supporting modeling and control of engineering systems – a case study,” In: *Proc. of the 14th International Conference on Intelligent Engineering Systems (INES)*, Las Palmas, Spain, pp. 117–122, 2010, ISBN: 978-1-4244-7650-3.

F3. T. A. Várkonyi, “Situation dependant evaluation of regression-type signal processing problems,” In: *Proc. of the 4th International Workshop on Soft Computing Applications (SOFA)*, Arad, Romania, pp. 225–228, 2010, ISBN: 978-1-4244-7985-6.

F4. A. R. Várkonyi-Kóczy, T. A. Várkonyi, “Anytime Model Regression,” In: *Proc. of the 10th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl’any, Slovakia, pp. 253–258, 2012, ISBN: 978-1-4577-0196-2.

References

[1] D. Pruessmann, B. Krause, C. von Altrock, “Fuzzy Logic Supervisory Control for Coal Power Plant,” In: *Proc. of the 6th IEEE International Conference on Fuzzy Systems*, 2, Barcelona, Spain, pp. 921–927, 1997.

[2] S. Yasunobu, S. Miyamoto, H. Ihara, “A Fuzzy Control for Train Automatic Stop Control,” *Transactions of the Society of Instrument and Control Engineers*, 19(11), pp. 873–880, 1983 (in Japanese); *Transactions of the Society of Instrument and Control Engineers*, E-2(1), pp. 1–9, 2002 (in English).

[3] S. Mahan, “Self-Driving Car Test,” `http://www.google.com/about/jobs/lifeatgoogle/self-driving-car-test-steve-mahan.html`.

[4] Y. Zhu, “Adaptive Output Feedback Control of Nonlinear Systems,” PhD thesis, 2001.

[5] J. J. Slotine, W. Li, “Applied Nonlinear Control,” Upper Saddle River: Prentice-Hall, 1991.

[6] V. I. Utkin, “Sliding Modes in Control and Optimization,” New York: Springer-Verlag, 1992.

[7] P. Korondi, “Tensor Product Model Transformation-based Sliding Surface Design,” *Acta Polytechnika Hungarica*, 3(4), pp. 23–35, 2006.

[8] K. Tanaka, H. O. Wang, “Fuzzy Control Systems Design and Analysis,” New York: John Wiley & Sons, Inc., 2001.

[9] L. X. Wang, “Adaptive Fuzzy Systems and Control: Design and Stability Analysis,” Upper Saddle River: Prentice-Hall, New Jersey, USA, 1994.

[10] A. R. Várkonyi-Kóczy, T. Kovácsházy, O. Takács, Cs. Benedecsik, “Anytime Algorithms in Intelligent Measurement and Control,” In: *CD-ROM Proc. of the World Automation Congress (WAC)*, Maui, USA, p. ISIAC-156, 2000.

[11] A. R. Várkonyi-Kóczy, “State Dependant Anytime Control Methodology for Non-linear Systems,” *International Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, 12(2), pp. 198–205, 2008.

[12] S. Ge, T. Lee, C. Harris, “Adaptive Neural Network Control of Robotic Manipulators,” World Scientific, 1998.

[13] C. Lin, C. Lee, “Neural-Network-Based Fuzzy Logic Control and Decision System,” *IEEE Tran. on computers*, 40(12), pp. 1320–1336, 1991.

[14] W. Miller III, R. Sutton, P. Werbos, “Neural Networks for Control,” 1990.

[15] V. Piuri, “Dynamic Reallocation of Processes and System Dimensioning in Fault-Tolerant Control Systems,” In: *Proc. of the IEEE Instrumentation and Measurement Technology Conference (IMTC)*, pp. 752–757, 1993.

[16] V. Piuri, “Design of Fault-Tolerant Distributed Control Systems,” *IEEE Transactions on Instrumentation and Measurement*, 43(2), pp. 257–264, 1994.

[17] Z. Qu, “Robust Control of Nonlinear Uncertain Systems,” Wiley Inc., New-York, 1998.

[18] K. Tanaka, T. Taniguchi, H.O. Wang, “Robust and Optimal Fuzzy Control: A Linear Matrix Inequality Approach,” In: *Proc. of the IFAC World Congress*, Beijing, pp. 213–218, 1999.

[19] W. Lin, C. Qian, “Adaptive Control of Nonlinearly Parameterized Systems: a Nonsmooth Feedback Framework,” *IEEE Trans. Automat. Contr.*, 47(5), pp.757–774, May 2002.

[20] J. P. Barbot, W. Perruquetti, “Sliding Mode Control in Engineering,” New York: Marcel Dekker, 2002.

[21] J. K. Tar, I. J. Rudas, K. R. Kozłowski, “Fixed Point Transformations-Based Approach in Adaptive Control of Smooth Systems,” *Lecture Notes in Control and Information Sciences 360 (Eds.: M. Thoma and M. Morari), Robot Motion and Control 2007 (Ed.: Krzysztof R. Kozłowski)*, Springer Verlag London Ltd., pp. 157–166, 2007.

[22] J. K. Tar, J. F. Bitó, I. J. Rudas, K. R. Kozłowski, J.A. Tenreiro Machado, “Possible Adaptive Control by Tangent Hyperbolic Fixed Point Transformations Used for Controlling the Φ^6-Type Van der Pol Oscillator,” In: *Proc. of the 6th IEEE International Conference on Computational Cybernetics (ICCC)*, Stará Lesná, Slovakia, pp. 15–20, 2008.

[23] J. K. Tar, J. F. Bitó, L. Nádai, J.A. Tenreiro Machado, “Robust Fixed Point Transformations in Adaptive Control Using Local Basin of Attraction,” *Acta Polytechnica Hungarica*, 6(1), pp. 21–37, 2009.

[24] J. K. Tar, J. F. Bitó, I. J. Rudas, “Replacement of Lyapunov’s Direct Method in Model Reference Adaptive Control with Robust Fixed Point Transformations,” In: *Proc. of the 14th IEEE International Conference on Intelligent Engineering Systems (INES)*, Las Palmas of Gran Canaria, Spain, pp. 231–235, 2010.

[25] J. K. Tar, I. J. Rudas, J. F. Bitó, K. R. Kozłowski, C. Pozna, “A Novel Approach to the Model Reference Adaptive Control of MIMO Systems,” In: *Proc. of the 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, Budapest, Hungary, pp. 31–36, 2010.

[26] A.M. Lyapunov, “A General Task About the Stability of Motion” (in Russian), PhD Thesis, 1892

REFERENCES

[27] A.M. Lyapunov, "Stability of Motion," Academic Press, New-York and London, 1966.

[28] P. A. Ioannou, J. Sun, "Robust Adaptive Control," Upper Saddle River: Prentice-Hall, New Jersey, USA, 1996.

[29] S. Bennett, "A History of Control Engineering: 1930-1955," Institution of Electrical Engineers, New York, USA, 1993.

[30] J. G. Ziegler, N. B. Nichols, "Optimum Settings for Automatic Controllers," *Transactions of the ASME*, 64, pp. 759–768, 1942.

[31] B. G. Lipták, "Instrument Engineers' Handbook: Process Control and Optimization," Boca Raton: CRC Press, 2006.

[32] L. R. Hunt, R. Su, G. Meyer, "Global Transformations of Nonlinear Systems," *IEEE Transactions on Automatic Control*, 28(1), pp. 24-31, 1983.

[33] B. Armstrong, O, Khatib, J. Burdick, "The Explicit Dynamic Model and Internal Parameters of the PUMA 560 Arm," In: *Proc. of the IEEE Conference On Robotics and Automation*, pp. 510–518, 1986.

[34] I. D. Landau, R. Lozano, M. M'Saad, A. Karimi, "Adaptive Control," London: Springer-Verlag, 2011.

[35] H. P. Whitaker, J. Yamron, A. Kezer, "Design of Model Reference Adaptive Control Systems for Aircraft," Report R-164, Instrumentation Laboratory, MIT, Cambridge, Massachussetts, 1958.

[36] R. L. Butchart, B. Shakcloth, "Synthesis of Model Reference Adaptive Control Systems by Lyapunovs Second Method," In: *Proc. 2nd IFAC Symposium on Theory of Self-adaptive Control Systems*, Plenum Press, Teddington, pp 145152, 1966.

[37] P. C. Parks, "Lyapunov Redesign of Model Reference Adaptive Control Systems," *IEEE Transactions on Automatic Control*, 11(3), pp. 362–367, 1966.

[38] "Model Reference Adaptive Control (MRAC)," `http://www.pages.drexel.edu/~kws23/tutorials/MRAC/MRAC.html`

[39] L.A. Zadeh, "Fuzzy Sets," *Information Control*, 8, pp. 338–353, 1965.

[40] A. R. Várkonyi-Kóczy, G. Péceli, T. P. Dobrowiecki, T. Kovácsházy, "Iterative Fuzzy Model Inversion," In: *Proc. of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Anchorage, Alaska, USA, 1, pp. 561–566, 1998.

[41] A. R. Várkonyi-Kóczy, "Model Based Anytime Soft Computing Approaches in Engineering Applications," In: *Balas, V., J. Fodor, A. R. Várkonyi-Kóczy (Eds.) Soft Computing Based Modeling in Intelligent Systems (Ser. Studies in Computational Intelligence)*, Springer Verlag, Berlin, Heidelberg, pp. 63-92, 2009.

[42] M. A. Khanesar, M. Teshnehlab, O. Kaynak, "Model Reference Fuzzy Control of Nonlinear Dynamical Systems Using an Optimal Observer," *Acta Polytechnica Hungarica*, 8(4), pp. 35-54, 2011.

[43] D. Tikk, L. T. Kóczy, T. D. Gedeon, "A survey on the universal approximation and its limits in soft computing techniques," Research Working Paper RWP-IT-01-2001, School of Information Technology, Murdoch University, Perth, W.A., 2001.

[44] T. Procyk, E. Mandami, "A Linguistic Self-organizing Process Controller," *Automatica*, 15(1), pp. 15–30, 1979.

[45] J. R. Layne, K. M. Passino, S. Yurkovich, "Fuzzy Learning Control for Anti-skid Braking Systems," *IEEE Trans. Control Systems Tech.*, 1(2), pp. 122–129, 1993.

[46] J. R. Layne, K. M. Passino, "Fuzzy Model Reference Learning Control for Cargo Ship Steering," *IEEE Control Systems Magazine*, 13(6), pp. 23-34, 1993.

[47] Zs. Cs. Johanyák, Sz. Kovács, "Polar-Cut Based Fuzzy Model for Petrophysical Properties Prediction," *Transactions on Automatic Control and Computer Science*, 57(67), pp. 195–200, 2008.

[48] S. Weber, "A General Concept of Fuzzy Connectives, Negations and Implications Based on T-norms and T-conorms," *Fuzzy Sets and Systems*, 11(1-3), pp. 103–113, 1983.

[49] A. Rövid, "Baleseti jármütest-deformációk identifikációja intelligens számítási módszerekkel," (Identification of Car-Body Crash Deformation Using Computational Intelligence Methods) PhD Thesis, Budapest University of Technology and Economics, 2005 (In Hungarian).

[50] P. J. Werbos, "An Overview of Neural Networks for Control," *IEEE Control Systems*, 11(1), pp. 40–41, 1991.

[51] F. Rosenblatt, "Principles of neurodynamics; perceptrons and the theory of brain mechanisms," Washington: Spartan Books, 1962.

[52] A. L. Hodgkin, A. F. Huxley, "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve," *The Journal of physiology*, 117(4), pp. 500–544, 1952.

[53] R. FitzHugh, "Impulses and Physiological States in Theoretical Models of Nerve Membrane," *Biophysical Journal*, 1(6), pp. 445–466, 1961.

[54] Y.-Q. Che, J. Wang, W.–L. Chan, K.–M. Tsang, "Chaos Synchronization of Coupled Neurons Under Electrical Stimulation via Robust Adaptive Fuzzy Control," *Nonlinear Dynamics*, 61(4), pp. 847–857, 2010.

[55] L. O. Chua, G. Lin, "Canonical Realization of Chua's Circuit Family," *IEEE Transactions on Circuits and Systems*, 37(7), pp. 885–902, 1990.

[56] T. Matsumoto, "A Chaotic Attractor from Chua's Circuit," *IEEE Transactions on Circuits and Systems*, 31(12), pp. 1055–1058, 1984.

[57] G. Duffing, "Erzwungene Schwingungen bei Veränderlicher Eigenfrequenz und ihre Technische Bedeutung," (Forced oscillations with variable natural frequency and their technical significance) Series: Sammlung Vieweg, 41/42, Vieweg & Sohn, Braunschweig, 1918 (In German).

[58] B. van der Pol, "Forced Oscillations in a Circuit with Non-Linear Resistance (Reception with Reactive Triode)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science Ser.*, 7(3), pp. 65–80, 1927.

[59] F. M. Kakmeni, S. Bowong, C. Tchawoua, E. Kaptouom, "Chaos Control and Synchronization of a Φ^6 -Van der Pol Oscillator," *Physics Letters A*, 322(5-6), pp. 305–323, 2004.

[60] A. Bonfanti, F. Pepe, C. Samori, A. L. Lacaita, "Flicker Noise Up-Conversion due to Harmonic Distortion in Van der Pol CMOS Oscillators," *IEEE Transactions on Circuits and Systems I*, 59(7), pp. 1418–1430, 2012.

[61] Igor M. Filanovsky, Luis B. Oliveira, "Regenerative Current Amplifier Using van der Pol Approximation," In: *54th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Seoul, Korea, pp. 1–4, 2011.

[62] David Kadjo, "A Fuzzy Classifier for an Inverted Pendulum," http://faculty.utep.edu/Portals/1255/david.pdf, 2006.

[63] S. P. Hoogendoorn, P. H. L. Bovy, "State-of-the-art of Vehicular Traffic Flow Modelling," In: *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4), pp. 283–303, 2001.

[64] B. D. Greenshields, "A Study of Traffic Capacity," *Highway Research Board Proceedings*, 14, pp. 448–477, 1935.

[65] M. Papageorgiou, "Multilayer Control System Design Applied to Freeway Traffic," *IEEE Transactions on Automatic Control*, 29(6), pp. 482–490, 1984.

[66] "Féktáv mérések," (Braking Distance Measurements) http://www.timeslip.hu/fektav.php (In Hungarian).

[67] E. Bakker, H.B. Pacejka, L. Lidner, "A New Tyre Model with an Application in Vehicle Dynamics Studies," *SAE paper 890087*, 1989.

[68] H.B. Pacejka, E. Bakker, "The Magic Formula Tyre Model," Tyre Models for Vehicle Dynamics (Hans B. Pacejka Ed.), In: *Proc. of* 1^{st} *International Colloquium on Tyre Models for Vehicle Dynamics Analysis, Delft, The Netherlands, 1991, (Supplement to Vehicle System Dynamics, Volume 21)* issued by SWETS & ZEITLINGER B.V. AMSTERDAM / LISSE, pp. 1–18, 1993.

[69] JJ. M. van Oosten, E. Bakker, "Determination of Magic Tyre Model Parameters," Tyre Models for Vehicle Dynamics (Hans B. Pacejka Ed.), In: *Proc. of the* 1^{st} *International Colloquium on Tyre Models for Vehicle Dynamics Analysis, Delft, The Netherlands, October 21–22, 1991, (Supplement to Vehicle System Dynamics, Volume 21)* issued by SWETS & ZEITLINGER B.V. AMSTERDAM / LISSE, pp. 19–29, 1993.

[70] L. Lidner, "Experience with the Magic Formula Tyre Model," Tyre Models for Vehicle Dynamics (Hans B. Pacejka Ed.), In: *Proc. of the* 1^{st} *International Colloquium on Tyre Models for Vehicle Dynamics Analysis, Delft, The Netherlands, October 21–22, 1991, (Supplement to Vehicle System Dynamics, Volume 21)* issued by Swets & Zeitlinger B. V. Amsterdam / Lisse, pp. 30–46, 1993.

[71] J. Lacombe, "Tire model for simulations of vehicle motion on high and low friction road surfaces," In: *Proc. of the* 32^{nd} *Conference on Winter Simulation Orlando, Florida, US*, pp. 1025–34, 2000.

[72] M. Burckhardt, "Fahrwerktechnik: Radschlupf–Regelsysteme," (Undercarriage: Wheel slip – Control Systems) Vogel-Verlag Würtzburg, Germany, 1993 (In German).

[73] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," (About the Operations in the Abstract Sets and Their Application to Integral Equations) *Fund. Math.*, 3, pp. 133–181, 1922 (In French).

[74] O. E. Rössler, "An Equation for Continuous Chaos," *Physics Letters A*, 57(5), pp. 397–398, 1976.

[75] E. N. Lorenz, "Deterministic Non-Periodic Flow," *Journal of the Atmospheric Sciences*, 20(2), pp. 130–141, 1963.

[76] Homepage of Scilab: http://www.scilab.org.

[77] A. Longtin, "Effects of Noise on Nonlinear Dynamics," In: *A. Beuter, L. Glass, M. C. Mackey, and M. S. Titcombe (Eds.) Nonlinear Dynamics in Physiology and Medicine*, Springer-Verlag, New York, USA, pp. 149–189, 2003.

[78] H. Kushner, "Filtering and Control for Wide Bandwidth Noise Driven Systems," *IEEE Transactions on Automatic Control*, 32(2), pp. 123–133, 1987.

[79] S. M. Kuo, "Nonlinear Adaptive Bilinear Filters for Active Noise Control Systems," *IEEE Transactions on Circuits and Systems I*, 52(3), pp. 617–624, 2005.

[80] D. Zhou, "Efficient Adaptive Nonlinear Filters for Nonlinear Active Noise Control," *IEEE Transactions on Circuits and Systems I*, 54(3), pp. 669–681, 2007.

[81] K. Gulez, "Torque Ripple and EMI Noise Minimization in PMSM Using Active Filter Topology and Field-Oriented Control," *IEEE Transactions on Industrial Electronics*, 55(1), pp. 251–257, 2008.

[82] B. Panomruttanarug, "Using Kalman filter to attenuate noise in learning and repetitive control can easily degrade performance," *In: Proc. of the SICE Annual Conference*, Tokyo, Japan, pp. 3453–3458, 2008.

[83] E. Ardestanizadeh, "Control-Theoretic Approach to Communication With Feedback," *IEEE Transactions on Automatic Control*, 57(10), pp. 2576–2587, 2012.

[84] C. Baylis, "Small Perturbation Harmonic Coupling in Nonlinear Periodicity Preservation Circuits," *IEEE Transactions on Circuits and Systems I*, 59(12), pp. 3034–3045, 2012.

[85] R. E. Kálmán, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, 82(1), pp. 35–45, 1960.

[86] R. E. Kálmán, R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Transactions of the ASME–Journal of Basic Engineering*, 83(3), pp. 95–108, 1961.

REFERENCES

[87] R. E. Kálmán, “Linear Stochastic Filtering Theory-Reappraisal and Outlook,” In: *Proc. of the Symposium on System Theory*, New York, USA, pp. 197–205, 1965.

[88] R. E. Kálmán, “System Identification from Noisy Data,” In: *A. Bednarek and L. Cesari (Eds.) Dynamical systems, II*, Academic Press, New York-London, pp. 135–164, 1982.

[89] R. E. Kálmán, “Zajos Rendszerek Identifikációja,” (Identification of Systems with Noise) *Alkalmazott Matemaikai Lapok*, 13(1-2), pp. 171–185, 1988 (In Hungarian).

[90] J. Zhang, Z. Zhang, “Application of a Strong Tracking Finite-Difference Extended Kalman Filter to Eye Tracking,” In: *D:-E. Huang, K. Li, G. W. Irwin (Eds.) Intelligent Computing, Book series Lecture Notes in Computer Science*, Springer-Verlag Berlin, Heidelberg, Germany, pp. 1170–1179, 2006.

[91] MathWorks documentation Center, “ode45,” `http://www.mathworks.com/help/matlab/ref/ode45.html`.

[92] J. K. Tar, “Towards Replacing Lyapunov’s “Direct” Method in Adaptive Control of Nonlinear Systems,” (invited plenary lecture), In: *Proc. of the 3rd Conference in Mathematical Methods in Engineering*, Coimbra, Portugal, Paper 11 (CD issue), 2010.

[93] G. J. Klir, B. Yuan, “Fuzzy Sets and Fuzzy Logic: Theory and Applications,” Upper Saddle River: Prentice Hall, 1995.

[94] P. Korondi, P. Bauer, P. J. van Duijsen, “Efficient Integrated Control and Circuit Simulation for Chattering Phenomena of a Motion Control System in Sliding Mode,” *International Review of Electrical Engineering*, 2(1), pp. 125–131, 2007.

[95] P. Guarneri, G. Rocca, M. Gob, “A Neural-Network-Based Model for the Dynamic Simulation of the Tire/Suspension System While Traversing Road Irregularities,” *IEEE Transaction on Neural Network*, 19(9), pp. 1549–1563, 2008.

[96] M. Madić, D. Marković, M. Radovanović, “Performance Comparison of Meta-Heuristic Algorithms for Training Artificial Neural Networks in Modelling Laser Cutting,” *International Journal of Advanced Intelligence Paradigms*, 4(3/4), pp. 299–312, 2012.

[97] H. Rasouli, C. Rasouli, A. Koohi, “Identification and Control of Plasma Vertical Position Using Neural Network in Damavand Tokamak,” *Review of Scientific Instruments*, 84(2), p. 023504–023504-12, 2013.

[98] A. R. Várkonyi-Kóczy, T. Kovácsházy, O. Takács, Cs. Benedecsik, “Anytime Algorithms in Intelligent Measurement and Control,” In: *CD-ROM Proc. of the World Automation Congress (WAC)*, Maui, USA, pp. ISIAC-156.1-6, 2000.

[99] A. R. Várkonyi-Kóczy, A. Rövid, “Observer Based Iterative Neural Network Model Inversion,” In: *Proc. of the 14th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Reno, USA, pp. 402–407, 2005.

[100] P. Korondi, A. R. Várkonyi-Kóczy, Sz. Kovács, P. Baranyi, M. Sugiyama,“Virtual Training of Potential Function Based Guiding Styles’, In: *Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference (IFSA / NAFIPS)*, Vancouver, Canada, pp. 2529–2534, 2001.

[101] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, N. Mastorakis, “Multilayer Perceptron and Neural Networks,” *WSEAS Transactions on Circuits and Systems*, 8(7), pp. 579–588, 2009.

[102] I. Filip, O. Prostean, V. E. Balas, G. Prostean, “Design and Simulation of a Neural Controller for Excitation Control of a Synchronous Generator,” In: *Proceedings of the 6th International Conference on Recent Advances in Soft Computing (RASC)*, Canterbury, UK, pp. 361–366, 2006.

[103] D. E. Rumelhart, G. E. Hinton, R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, 323(6088), pp. 533–536, 1986.

[104] J. J. Hopfiled, “Neural Networks and Physical Systems with Emergent Collective Computational Abilities,” In: *Proceedings of the National Academy of Sciences of the United States of America*, National Academy Sciences, 79(8), pp. 2554–2558, 1982.

[105] K. Souissi, F. Odeh, H. H. K. Tang, A. Gnudi, Pong-Fei Lu, “Investigation of the Impact Ionization in the Hydrodynamic Model,” *IEEE Transactions on Electron Devices*, 40(8), pp. 1501–1507, 1993.

[106] M. A. Alsunaidi, S. M. El- Ghazaly, “High-Frequency Time-Domain Modeling of GaAs FETs Using Hydrodynamic Model Coupled with Maxwell’s Equations,” In: *Proc. of the IEEE MTT-S International Microwave Symposium Digest*, San Diego, CA, USA, pp. 397–400, 1, 1994.

[107] J. A. Maruhn, “The Hydrodynamic Model for High-Energy Heavy-Ion Collisions,” *Il nuovo cimento A*, 87(2), pp. 125–138, 1985.

[108] Y. I. Londer, K. N. Ulyanov, “Model of the Short Vacuum Arc at Collision Free Motion of Ions ,” In: *Proc. of the 25th International Symposium on Discharges and Electrical Insulation in Vacuum (ISDEIV)*, Tomsk, Russia, pp. 341–344, 2012.

[109] T. Luspaya, B. Kulcsár, I. Varga, J. Bokor, “Parameter-Dependent Modeling of Freeway Traffic Flow,” *Transportation Research Part C: Emerging Technologies*, 18(4), pp. 471–488, 2009.

[110] H.B. Callen, “Thermodynamics and an Introduction to Thermostatistics,” 2nd Edition, John Wiley & Sons Inc., 1985.

[111] D. Kondepudi, I. Prigogine, “Modern Thermodynamics,” John Wiley & Sons, Chichester, 1998.

[112] L. Magni, D. M. Raimondo, C. Dalla Man, G. De Nicolao, B. Kovatchev, C. Cobelli, “Model Predictive Control of Glucose Concentration in Type I Diabetic Patients: An In Silico Trial,” *Biomedical Signal Processing and Control*, 4(4), pp. 338–346, 2009.

[113] C. Dalla Man, R. Rizza, C. Cobelli, "Meal Simulation Model of the Glucose-Insulin System," *IEEE Transactions in Biomedical Engineering*, 54(10), pp. 1740–1749, 2007.

[114] J. Abadie, J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," In: *R. Fletcher (Ed.) Optimization*, London: Academic press, 1969.

[115] M. Papageorgiou, A. Kotsialos, "Freeway Ramp Metering: An Overview," In: *IEEE Intelligent Transportation Systems Conference Proceedings Dearborn (MI)*, USA, 2000.

[116] "Traffic Detector Handbook," Third Edition Vol. 1 Research, Development, and Technology Turner-Fairbank Highway Research Center, publication no. FHWA-HRT-06-108, 2006.

[117] I. J. Rudas, J. K. Tar, L. Nádai, "System and Control Theory with Especial Emphasis on Nonlinear Sytems," Budapest, Hungary: Typotex Elektronikus Kiad´, 2012.

[118] B. Armstrong-Hèlouvry, "Stick Slip and Control in Low Speed Motion," *IEEE Trans. On Automatic Control*, 38(10), pp. 1483–1496, 1990.

[119] L. Márton, B. Lantos, "Friction and Backlash Induced Limit Cycles in Mechanical Control Systems," In: *Proc. of the European Control Conference ECC*, Budapest, Hungary, pp. 23–26, 2009.

[120] A. Zanten, R. Erhardt, A. Lutz, "Measurement and Simulation of Transients in Longitudinal and Lateral Tire Forces," *SAE Paper 900210*, 99(6), pp. 300–318, 1990.

[121] C. Canudas de Wit, P. Tsiotras, "Dynamic Tire Friction Models for Vehicle Traction Control," In: *Proc. of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, USA, pp. 3746–3751, 1999.

[122] B. Olson, "Nonlinear Dynamics of Longitudinal Ground Vehicle Traction," MS thesis, Michigan State University, East Lansing, MI, 2001.

[123] M. Bian, K. Lee, N. Feng, "An Empirical Model for Longitudinal Tire–Road Friction Estimation," In: *Proc. of the SAE 2004 World Congress & Exhibition*, Detroit, MI, USA, pp. 10, 2004.

[124] J. Svendenius, "The Tire Models for Use in Braking Applications," Department of Automatic Control, Lund Institute of Technology, Sweden, 2003.

[125] W.-E. Ting, J.-S. Lin, "Nonlinear Control Design of Anti-Lock Braking Systems Combined with Active Suspensions," In: *Proceedings of the* 5th *Asian Control Conference*, Melbourne, Australia, 2004.

[126] S. Anwar, B. Zheng, "An Antilock-Braking Algorithm for an Eddy-Current-Based Brake-by-Wire System," *IEEE Trans. on Vehicular Technology*, 56(3), pp. 1100–1107, 2007.

[127] H. S. Tan, M. Tomizuka, "An Adaptive Sliding Mode Vehicle Traction Controller Design," In: *Proc. of the American Control Conference*, Pittsburgh, PA, USA, pp. 1856–1861, 1989.

[128] Y. K. Chin, W. C. Lin, D. M. Sidlosky, M. S. Sparschu, "Sliding-Mode ABS Wheel Slip Control," In: *Proc. of the American Control Conference*, Chicago, IL, USA, pp. 1–6, 1992.

[129] S. Drakunov, U. Ozguner, P. Dix, B. Ashrafi, "ABS Control Using Optimum Search via Sliding Modes," *IEEE Transactions on Control Systems Technology*, 3(1), pp. 79–85, 1995.

[130] P. Kachroo and M. Tomizuka, "Sliding Mode Control with Chattering Reduction and Error Convergence for a Class of Discrete Nonlinear Systems with Application to Vehicle Control," In: *Proc. of the International Mechanical Engineering Congress and Expo*, 57(1), pp. 225–233, 1995.

[131] C. Unsal and P. Kachroo, "Sliding Mode Measurement Feedback Control for Antilock Braking Systems," *IEEE Transactions on Control Systems Technology*, 7(2), pp. 271–281, 1999.

[132] A. Harifi, A. Aghagolzadeh, G. Alizadeh, M. Sadeghi, "Designing a Sliding Mode Controller for Antilock Brake System," In: *Proc. of The International Conference on Computer as a Tool*, Serbia and Montenegro, pp. 611–616, 2005.

[133] J. R. Layne, K. M. Passino, S. Yurkovich, "Fuzzy Learning Control for Antiskid Braking Systems," *IEEE Transaction on Control System Technology*, 1(2), pp. 122–129, 1993.

[134] G. F. Mauer, "A Fuzzy Logic Controller for an ABS Braking System," *IEEE Transactions on Fuzzy Systems*, 3(4), pp. 381–388, 1995.

[135] Y. Lee, H. S. Zak, "Genetic Neural Fuzzy Control of Anti-Lock Brake Systems," In: *Proc. of the 2001 American Control Conference*, Arlington, VA, USA, Vol. 2, pp. 671–676, 2001.

[136] C. M. Lin, C.F. Hsu, "Self-Learning Fuzzy Sliding-Mode Control for Antilock Braking Systems," *IEEE Transactions on Control Systems Technology*, 11(2), pp. 273–278, 2003.

[137] A. V. Topalov, E. Kayacan, O. Kaynak, "Neuro–Fuzzy Control of Antilock Braking System Using Variable-Structure-Systems-based Learning Algorithm," In: *Proc. of the International Conference on Adaptive and Intelligent Systems (ICAIS)*, Klagenfurt, Austria, pp. 166–171, 2009.

[138] Y. Oniz, E. Kayacan, O. Kaynak, "A Grey Sliding Mode Controller Design for Antilock Braking System," In: *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2007)*, Las Vegas, Nevada, USA, pp. 267–274, 2007.

[139] S. John, J. O. Pedro, L. T. Kóczy, "Adaptive Improvement of a Passive Antilock Brake Control," In: *Proc. of IEEE AFRICON*, Livingstone, Zambia, pp. 1–6, 2011.

[140] J. Richalet, A. Rault, J. L. Testud, J. Papon, "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica*, 14(5), pp. 413–428, 1978.

Appendix A

Acronyms

CD	Compact Disk
RFPT	Robust Fixed Point Transformations
CTC	Computed Torque Control
PID	Proportional-Integral-Derivative
PD	Proportional-Derivative
PI	Proportional-Integral
MRAC	Model Reference Adaptive Controller
SC	Soft Computing
FLC	Fuzzy Logic Controller
CoA	Center of Area defuzzification method
CoG	Center of Gravity defuzzification method
CoM	Center of Maxima defuzzification method
MoM	Mean of Maxima defuzzification method
NN	Neural network
NNC	Neural network controller
MLP	MultiLayer Perceptron
LMS	Least Mean Square algorithm
FHN	FitzHugh-Nagumo
SMC	Sliding Mode Controller
GRG	Generalized Reduced Gradient
MS	Microsoft
ABS	Anti-lock Braking System

Appendix B

List of notations

α	Free variable of the Φ^6-type Van der Pol oscillator
α_1	Free variable of the master Duffing system
α_2	Free variable of the slave Duffing system
β	Free parameter of a noise filter
β_1	Free variable of the master Duffing system
β_2	Free variable of the slave Duffing system
$\cdot$	Symbol that marks differentiated value (one dot: first derivative; two dots: second derivative)
Δt_{Cycle}	Cycle time of RFPT
$\Delta t_{sampling}$	Sampling time of the sensors in the hydrodynamical model of freeway traffic
Δ	The step between the A_i values
δ	Free parameter of the hydrodynamical model of freeway traffic
δ_1	Free variable of the master Duffing system
δ_2	Free variable of the slave Duffing system
η	Free parameter of the hydrodynamical model of freeway traffic
$\hat{\rho}_0$	The stationary number of the vehicles in the zeroth road segment
$\hat{\rho}_1$	The stationary number of the vehicles in the first road segment
$\hat{\rho}_2$	The stationary number of the vehicles in the second road segment
$\hat{\rho}_3$	The stationary number of the vehicles in the third road segment
$\hat{\rho}_4$	The stationary number of the vehicles in the fourth road segment
$\hat{\rho}_5$	The stationary number of the vehicles in the fifth road segment
$\hat{\tau}_{abs}$	Approximated time constant of the hydraulic braking system
$\hat{a}$	Free parameter of the approximate FitzHugh-Nagumo neuron model
$\hat{b}$	Free parameter of the approximate FitzHugh-Nagumo neuron model
$\hat{B}_v$	Approximated friction coefficient
$\hat{B}_w$	Approximated friction coefficient
$\hat{b}_{cp}$	Approximated friction in the cart-pendulum model
$\hat{C}_1^{mc}$	First capacitor of the approximate Matsumoto-Chua model
$\hat{C}_2^{mc}$	Second capacitor of the approximate Matsumoto-Chua model
$\hat{G}$	Reciprocal value of a common resistor of the approximate Matsumoto-Chua model
$\hat{g}$	Approximated gravity, $\hat{g} = 10\,m/s^2$
$\hat{g}_c$	Coupling element of the approximate FitzHugh-Nagumo neuron model
$\hat{k}$	Approximate parameter of the Φ^6-type Van der Pol model

$\hat{L}_1$	Approximated length of the first pendulum in the cart plus double pendulum model
$\hat{L}_2$	Approximated length of the second pendulum in the cart plus double pendulum model
$\hat{L}_c$	Inductance of a coil of the approximate Matsumoto-Chua model
$\hat{L}_p$	Approximated length of the pendulum in the cart-pendulum model
$\hat{M}$	Approximated mass of the cart in the cart-pendulum and the cart plus double pendulum models
$\hat{m}$	Approximated mass
$\hat{m}_1$	Approximated mass of the first pendulum in the cart plus double pendulum model
$\hat{m}_2$	Approximated mass of the second pendulum in the cart plus double pendulum model
$\hat{m}_{vdp}$	Approximate inertia of the Φ^6-type Van der Pol model
$\hat{r}_2$	Fixed ingress rate from the ramp in the second road segment
$\hat{S}_{big}$	Approximate parameter of the Matsumoto-Chua circuit
$\hat{S}_{small}$	Approximate parameter for determining the value of g_c
$\hat{v}_0$	Stationary velocity of the traffic in the zeroth road segment
$\hat{v}_1$	Stationary velocity of the traffic in the first road segment
$\hat{v}_2$	Stationary velocity of the traffic in the second road segment
$\hat{v}_3$	Stationary velocity of the traffic in the third road segment
$\hat{v}_4$	Stationary velocity of the traffic in the fourth road segment
$\hat{v}_5$	Stationary velocity of the traffic in the fifth road segment
κ	Free parameter of the hydrodynamical model of freeway traffic
λ	Free parameter of the PI, PD, and PID controllers
λ_t	Wheel slip
λ_{ft}	Number of lanes in the hydrodynamical model of freeway traffic
λ_{max}	Maximal wheel slip
λ_{vdp}	Free variable of the Φ^6-type Van der Pol oscillator
μ	Friction coefficient
μ_{vdp}	Free variable of the Φ^6-type Van der Pol oscillator
ω_0	Free variable of the Φ^6-type Van der Pol oscillator
ω	Rotational velocity
ω_d	Frequency of the excitation of the Duffing systems
$\overline{C}$	Average for the vehicles in the hydrodynamical model of freeway traffic
ρ_0	The number of the vehicles in the zeroth road segment
ρ_1	The number of the vehicles in the first road segment
ρ_2	The number of the vehicles in the second road segment
ρ_3	The number of the vehicles in the third road segment
ρ_4	The number of the vehicles in the fourth road segment
ρ_5	The number of the vehicles in the fifth road segment
ρ_{cr}	Critical vehicle density of the freeway traffic

τ	Free parameter of the hydrodynamical model of freeway traffic
τ_{abs}	Time constant of the hydraulic braking system
θ	Angle of the pendulum in the cart-pendulum system
θ_1	The angle of the first pendulum in the cart plus double pendulum system
θ_2	The angle of the second pendulum in the cart plus double pendulum system
$\tilde{\delta}$	Approximate parameter of the hydrodynamical model of freeway traffic
$\tilde{\eta}$	Approximate parameter of the hydrodynamical model of freeway traffic
$\tilde{\kappa}$	Approximate parameter of the hydrodynamical model of freeway traffic
$\tilde{\lambda}_{ft}$	Approximate number of lanes in the hydrodynamical model of freeway traffic
$\tilde{\rho}_{cr}$	Approximate critical vehicle density of the freeway traffic
$\tilde{\tau}$	Approximate parameter of the hydrodynamical model of freeway traffic
$\tilde{b}$	Approximate parameter of the Papageorgiou model
$\tilde{C}_1^{mc}$	First capacitor of the slave Matsumoto-Chua model
$\tilde{C}_2^{mc}$	Second apacitor of the slave Matsumoto-Chua model
$\tilde{G}$	Reciprocal value of a common resistor of the slave Matsumoto-Chua model
$\tilde{g}_{mc}$	Nonlinear element of the slave Matsumoto-Chua circuit
$\tilde{L}_c$	Inductance of a coil of the slave Matsumoto-Chua model
$\tilde{L}_{rs}$	Approximate length of the road segments
$\tilde{S}_{big}$	Free parameter for determining the value of $\tilde{g}_{mc}$ in the slave Matsumoto-Chua circuit
$\tilde{S}_{small}$	Free parameter for determining the value of $\tilde{g}_{mc}$ in the slave Matsumoto-Chua circuit
$\tilde{v}_{free}$	Approximate velocity of the traffic flow
φ	Representative of the controlled systems
φ_{appr}^{-1}	Representative of the approximate inverse models of the controlled systems
ς	The height of Gaussian-like function ϑ
ϑ	Gaussian-like function
ϑ^{Est}	Approximation of ϑ_{abs}
ϑ_{abs}	$\vartheta_{abs} := F_z\mu$
$\wedge$	Symbol that marks approximated value
ζ	The center of the Gaussian-like function ϑ
$\{A_i\}$	Possible values for parameter A
A	Free parameter of functions G_1, G_2, and G_3
a	Amplitude of the excitation of the Duffing systems
a_1	Free parameter of the FitzHugh-Nagumo neuron model
a_2	Free parameter of the FitzHugh-Nagumo neuron model

B	Free parameter of functions G_1, G_2, and G_3
b	Free parameter of the Papageorgiou model
B_1	Free variable of the Burckhardt tire model
b_1	Free parameter of the FitzHugh-Nagumo neuron model
B_2	Free variable of the Burckhardt tire model
b_2	Free parameter of the FitzHugh-Nagumo neuron model
B_3	Free variable of the Burckhardt tire model
B_4	Free variable of the Burckhardt tire model
B_v	Friction coefficient
B_w	Friction coefficient
b_{cp}	Friction in the cart-pendulum system
C_1^{mc}	First capacitor of the master Matsumoto-Chua circuit
C_2^{mc}	Second capacitor of the Matsumoto-Chua circuit
C_1	Proposed controller without RFPT extenstion
C_2	Proposed controller, usually with RFPT extenstion (except Chapter 6)
C_3	PD controller without RFPT extenstion in Chapter 6 or RFPT-based PID controller with parameter tuning in Chapter 7
C_4	RFPT-based "recalculated PD controller
C_5	RFPT-based model reference adaptive controller
C_6	RFPT-based PD controller
d_1	Disturbance force for the mater systems
D_ω^{Max}	Opposite of the maximal deceleration for ω
D_v^{Max}	Opposite of the maximal deceleration for the car body
e	Tracking error
e_1	Tracking error of the first state variables
e_2	Tracking error of the second state variables
E_f	Emission factor in the hydrodynamical model of freeway traffic
F	Applied force in the cart-pendulum system
f	$f\left(x\right) = \varphi\left(\varphi_{appr}^{-1}\left(x\right)\right)$, where $f(r_*) = r^d$
F_1	Applied force on the first pendulum in the cart plus double pendulum system
F_2	Applied force on the second pendulum in the cart plus double pendulum system
F_z	Vertical contact force
G	Reciprocal value of a common resistor of the master Matsumoto-Chua circuit
g	Gravity, $g = 9.81\,m/s^2$
$G\left(A_i\right)$	Function G_1, G_2, or G_3 where parameter A is replaced with A_i
G_1	The function used by the RFPT-based MRAC
G_2	The function used by the RFPT-based PD controller
G_3	The function used by the recalculated RFPT-based PD controller
g_c	Coupling element for two FitzHugh-Nagumo neurons

g_{mc}	Nonlinear element of the master Matsumoto-Chua circuit
H	The output of the fuzzy-like parameter tuning
h	$h\left(x\right) = \varphi_{appr}^{-1}\left(\varphi\left(x\right)\right)$, where $h(u^d) = u^d_{appr}$
I	External excitation of the systems causing chaotic behavior
i_L	Current of the inductance of the master Matsumoto-Chua circuit
i_u	Control current for the slave Matsumoto-Chua circuit
J	Constant inertial momentum of the wheel/wheel shaft/motor system
K	Free parameter of functions G_1, G_2, and G_3
K_b	Gain constant of the proposed vehicle model
K_{vssm}	Free parameter for the chattering reduction algorithm
L_1	Length of the first pendulum in the cart plus double pendulum system
L_2	Length of the second pendulum in the cart plus double pendulum system
L_c	Inductance of a coil of the master Matsumoto-Chua circuit
L_p	The length of the pendulum in the cart-pendulum system
L_{rs}	The length of the road segments
M	Mass of the cart in the cart-pendulum and the cart plus double pendulum system
m	Mass
m_1	Mass of the first pendulum in the cart plus double pendulum system
m_2	Mass of the second pendulum in the cart plus double pendulum system
m_{vdp}	Some inertia of the Φ^6-type Van der Pol oscillator
P_b	Pressure in the braking system
P_b^{Est}	Estimated pressure in the braking system
Q	Applied torque or control force
Q^d_{appr}	Approximate control force calculated by the approximate inverse model
q_0	The traffic current density in the zeroth road segment
q_1	The traffic current density in the first road segment
q_2	The traffic current density in the second road segment
q_3	The traffic current density in the third road segment
q_4	The traffic current density in the fourth road segment
q_5	The traffic current density in the fifth road segment
r	Radius of the wheel
r^*_G	System response so that $f(PD(r^*_G)) = r^d$
r^d	Desired system response
r^d_n	Desired system response in the n$^{\text{th}}$ simulation step
r^r	Realized system response in the n$^{\text{th}}$ simulation step
r^r	Realized system response
r_*	System response so that $\varphi_{appr}^{-1}\left(r_*\right) = u^d$
r_2	The ingress rate from the ramp in the second road segment
S_{big}	Free parameter for determining the value of g_{mc} in the master Matsumoto-Chua circuit

S_{small}	Free parameter for determining the value of g_{mc} in the master Matsumoto-Chua circuit
t	Time
T_b	Braking torque
T_b^{Est}	Estimated braking torque
t_n	Time of the n^{th} control step
u	Control force
u^d	Desired control force
u^d_{appr}	Approximate control signal calculated by the approximate inverse model
v	Velocity
v_0	The velocity of the traffic in the zeroth road segment
v_1	The velocity of the traffic in the first road segment
v_2	The velocity of the traffic in the second road segment
v_3	The velocity of the traffic in the third road segment
v_4	The velocity of the traffic in the fourth road segment
v_5	The velocity of the traffic in the fifth road segment
v_ζ	The speed of the movement of Gaussian-like function ϑ
v_{C1}	Voltage of capacitor C_1^{mc} of the master Matsumoto-Chua circuit
v_{C2}	Voltage of capacitor C_2^{mc} of the master Matsumoto-Chua circuit
v_{free}	Velocity of the traffic flow
w_i	Weighting factors for parameters A_i
x	System state variable
x^D	Desired state of the systems calculated by a controller
x^d	Desired state of the systems calculated by a controller
x^N	Marks the nominal trajectory for a system with state variable x
x^{Des}	Desired state of the systems calculated by a controller
x^{Nom}	Marks the nominal trajectory for a system with state variable x
x_1	Master system's first state variable
x_2	Mater system's second state variable
x_c	The position of the cart in the cart-pendulum and the cart plus double pendulum systems
y_1	Slave system's first state variable
y_2	Slave system's second state variable

Appendix C

List of figures

Appendix D

List of tables

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other Hungarian, Italian or foreign examination board.

The thesis work was conducted from 09/01/2010 to 03/31/2013 under the supervision of Professor József K. Tar at Óbuda University and Professor Vincenzo Piuri at Universitá degli Studi di Milano.

Budapest, 05/17/2013