Università degli Studi di Milano

A Thesis submitted for the degree of Doctor of Philosophy

NON-CONVENTIONAL DATA REPRESENTATION AND CONTROL

Adrienn Dineva

Supervisor:

Prof. Dr. Vincenzo Piuri

Co-Supervisors:

Prof. Dr. Annamária R. Várkonyi-Kóczy
Prof. Dr. József K. Tar

December 2016

Members of the Comprehensive Examination Committee:

Members of the Defense Committee:

Date of the defense:

Acknowledgements

First and foremost I would like to express my deepest gratitude to my dear Supervisor Prof. Vincenzo Piuri and co-Supervisors Prof. Annamária R. Várkonyi-Kóczy and Prof. József K. Tar for their generous support and guidance.

I am grateful for the continuous support by Prof. Vincenzo Piuri in my PhD study. I highly appreciate his enthusiasm and expertise that allowed me to undertake this research

I am grateful for the ideas and time of Prof. Annamária R. Várkonyi-Kóczy with which she supported me to complete my PhD Thesis.

I especially acknowledge the guidance by Prof. József K. Tar. His patience, inspiration, and immense knowledge helped me a lot in the completion of this Thesis.

I would like to sincerely thank Prof. József Dombi for the interesting discussions and his suggestions and support for completing my research.

I gratefully acknowledge the support of the Doctorate School of the Ph.D. Degree Program in Computer Science, and the Department of Computer Science, Università degli Studi di Milano.

Lastly, I would like to express my thanks to my Family for all their love and support.

Contents

1 Introduction **7**

1.1 Research Aims and Their Relevance in the Context of the State of the Art . 7

1.2 Organization of the Thesis and Main Directions 9

1.3 Research Methodology 12

2 Combination of Classical Model Identification with the RFPT-based Design by the Use of a New Tuning Method **13**

2.1 Principles of the Original Robust Fixed Point Transformation for Nonlinear Control 16

2.2 Critical Analysis and Modification of the AIDC Controller 17

2.2.1 The Operation of the Classical AIDC Controller 17

2.2.2 Modified Tuning Algorithm 19

2.3 Combination of the New Tuning Method with the RFPT-based Adaptive Control 24

2.3.1 Simulation Results for the RFPT-supported AIDC Controller with Modified Tuning Rule 25

2.4 Combination of the Modified Adaptive Slotine-Li Robot Controller (AD-SLRC) with the RFPT-based Adaptive Controller 28

2.4.1 The Tuning Method using Lyapunov-function 28

2.4.2 New Parameter Tuning 30

2.4.3 Further Modification in the Exerted Force/Torque Components . . . 31

2.4.4 Simulation Results 31

2.4.5 Cooperation in the Lack of External Disturbances 32

2.4.6 Cooperation under the Effect of a LuGre Friction at Axle 3 32

2.5 Novel Tuning Method for the Modified Adaptive Inverse Dynamic Robot Controller (MAIDRC) 37

2.5.1 Improvement in the MAIDRC control design 37

2.5.2 Simulation Results 38

2.6 Thesis Statement I. 47

3 New Generation of Fixed Point Transformation for Adaptive Control 48

3.1 Fixed Point's Generation for SISO Systems 48

3.1.1 The Idea of Fixed Point Generation 48

3.1.2 Application Example 50

3.2 Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO Systems 58

3.2.1 The Extension to MIMO Systems 58

3.2.2 Application Example 61

3.3 New Advances Regarding the Parameter Tuning 66

3.3.1 Replacement of Parameter Tuning with Simple Calculation 66

3.3.2 Application Example 67

3.4 Thesis Statement II. 70

3.4.1 Substatement I. 70

3.4.2 Substatement II. 70

4 Advances in the Sigmoid Generated Fixed Point Transformation 71

4.1 Adaptive Control Using Improved Sigmoid Generated Fixed Point Transformation and Scheduling Strategy 71

4.1.1 New Function 72

4.1.2 The Control Design for Underactuated Mechanical Systems 72

4.1.2.1 Realization of the Suggested Control Method using Stretched Sigmoid Function 73

4.1.2.2 Simulation Results 73

4.2 Novel Type of Function 76

4.2.1 Validation of Practical Applicability Through the Adaptive Control of Kapitza's Pendulum System 77

4.2.1.1 Results of Numerical Simulations 78

4.3 Enhancement of the SGFPT Control Design by Soft Computing 82

4.3.1 The System under Consideration 82

4.3.2 The Control Strategy 83

4.3.3 Results for the Affine Model 84

4.3.4 Results for the Soft Computing-based Model 86

4.3.5 Results for the Fully Soft Computing-based Model 88

4.4 Thesis Statement III. 91

4.4.1 Substatement I. 91

4.4.2 Substatement II. 91

5 Improved Denoising in the Wavelet Domain 92

5.1 Wavelet Shrinkage 93

5.1.1 Fuzzy Supervisory System 94

5.1.2 Improved Denoising 96

5.1.3 Simulation Results 96

5.2 Thesis Statement IV. 100

6 Adaptive Multi-round Smoothing based on the Savitzky - Golay Filter 101

6.1 Brief Inroduction of the Mathematical Background behind the Savitzky-Golay Filter 102

6.2 Adaptive Multi-round Smoothing based-on the SG Filtering Technique . . . 104

6.2.1 Multi-round Smoothing and Correction by the use of Fuzzy Rules . 104

6.2.2 New Parametric Weighting Function 105

6.2.3 Simulation Results 107

6.3 Thesis Statement V. 109

7 Conclusions 110

8 Possible Targets of Future Research 113

9 References 122

Chapter 1

Introduction

Among the different approaches of modern engineering applications model-integrated computing plays an exceptional role. Modeling is a fundamental and difficult problem in all the sciences; to design a controller one needs a model. Soft Computing techniques, such as fuzzy and neural network-based models, are found to be highly efficient due to their flexibility, robustness and easy interpretability. Especially in cases where the problem to be solved is highly nonlinear or when only partial, uncertain and/or inaccurate data is available. At the same, though their usage can be so advantageous, it is still limited by their exponentially increasing computational complexity. Combining Soft Computing, non-conventional and novel data representation techniques is a possible way to overcome this difficulty. The performance of a controller depends on the available form of the model, therefore my research concentrates on novel data representation and control methods that are able to adaptively cope with usually imperfect, noisy or even missing information, the dynamically changing, possibly insufficient amount of resources and reaction time (for instance, wavelet based multiresolution controllers [1], anytime control [2][3][4][5], Situational Control [6][7], Robust Fixed Point Transformation-based control [8], etc.).

1.1 Research Aims and Their Relevance in the Context of the State of the Art

The field of $Adaptive\ Systems$, that includes for instance recursive identification, adaptive control, filtering, and signal processing, has been one of the most active research areas of the past decade [9][10]. Since adaptive controllers are fundamentally nonlinear, their theoretical analysis is usually very difficult. Therefore, modern approaches of control design

and signal processing include a various class of mathematical tools [5][11][12]. The idea of wavelet based controllers (see, [1][13][14][15]) originates from the facilities of series expansion with wavelets. In paper [1] the authors investigate wavelet network and fuzzy approximation in controlling a class of continuous time unknown nonlinear systems. The described method applies variable wavelet bases, where the adjustable parameters enable constructing suitable control laws. An effective way dealing with potentially infinite number of unknown parameters with the help of wavelet basis functions has been introduced in [13]. The proposed method is based on constructing an ideal infinite controller and approximating its behaviour with a finite controller. The authors highlight the advantages of the 'Mexican hat' type wavelet frames from multiresolution analysis' point of view. Paper [14] shows a new frequency-domain approach to identify poles in discrete-time linear systems. The discrete rational transfer function is represented in a rational Laguerre-basis, where the basis elements are expressed by powers of the Blaschke-function. This function can be interpreted as a congruence transform on the Poincaré unit disc model of the hyperbolic geometry. The identification of a pole is given as a hyperbolic transform of the limit of a quotient-sequence formed from the Laguerre-Fourier coefficients. Paper [14] extends this approach for using discrete time-domain data directly. Another interesting new adaptive fuzzy wavelet network controller is shown in [15], for control of nonlinear affine systems, inspired by the theory of multiresolution analysis (MRA) of wavelet transforms and fuzzy concepts. The proposed adaptive gain controller, which results from the direct adaptive approach, has the ability to tune the adaptation parameter in each fuzzy rule during real-time operation.

The traditional approach in the design of adaptive controllers for nonlinear dynamic systems normally applies Lyapunov's "direct" method [16]. Several solutions have been proposed in order to replace this technique by a simpler approach (see, [17][18][19][20][21][22]). The main characteristic features of Lyapunov's method can be summarized as follows: a) it yields satisfactory conditions for the stability, b) instead of focusing on the primary design intent (for instance, the precise prescription of the trajectory tracking error relaxation) it concentrates on proving "global stability" that often is too much for common practical applications, c) in the identification of the model parameters of the controlled system it provides a tuning algorithm that contains certain components of the particular Lyapunov function in use, therefore it works with a large number of arbitrary adaptive control parameters; (see, [22]), d) the parameter identification process in certain cases is vulnerable if unknown external perturbations can disturb the system under control. Concentrating on the primary design intent the "Robust Fixed Point Transformation (RFPT)"-based technique was suggested. The RFPT at the cost of sacrificing the need for global stability – applied iteratively deformed control signal sequences that, on the basis of Banach's Fixed Point Theorem, converged to the ap-

propriate control signal only within a bounded basin of attraction. This method was found to be applicable for a wide class of systems to be controlled, it was robust against the unknown external disturbances. Various tuning methods were suggested for keeping the control signal in the basin of attraction of the fixed point [23], later its global properties were investigated in [24], [25], [26], and [27]. These investigations resulted in the following conclusion: for a wide class of physical systems it has become always possible to so tune one of the adaptive control parameters, that so called "precursor oscillations" appear when the monotone convergent sequence turns into a non-monotone but still convergent one before turning into bounded chaotic fluctuations. Since it was possible to observe the precursor oscillations with a simple, model-independent observer, it also has became possible to maintain the convergence, therefore the lack of guaranteed global stability was efficiently compensated from the point of view of practical applications. The RFPT –based control has also been applied in various tasks, like in chaos synchronization [28] and traffic control [29]. As Lyapunov's Direct Method can be applied in the Model Reference Adaptive Control [22][30], the Robust Fixed Point Transformations can also be used for such purposes [31]. Further interesting results have been obtained in the control of certain dynamical systems (for example [32][33]). The above summarized antecedents and the preliminary results introduced in my M.Sc. Thesis [34] provided interesting prospects for further investigations. The main directions for this research will be outlined in the next section.

1.2 Organization of the Thesis and Main Directions

The deep theoretical side to control and signal processing is ubiquitous in any system, whether it be mechanical or electrical. This theoretical side provides a systematic approach to the design of control and signal processing algorithms for practical engineering problems. Therefore, more sophisticated algorithms are required in adaptive systems. This research makes an attempt to introduce new algorithmical methods to adaptive control in the first three theses and to adaptive signal procesing in the last two theses. The main objectives are detailed below:

- Chapter 2 aims revealing the possibilities of the combination of classical model-identification and the RFPT-based design. The proposed new method utilizes the geometric interpretation provided by the Lyapunov-technique that can be directly used for parameter tuning. It is shown that these useful information are obtained by using the same feedback terms and equations of motion, as in the original method. The application of the modified Gram-Schmidt algorithm is proposed for the new parameter tuning strategy with the appropriate modifications of the "*Adaptive Inverse Dynamics Controller*

(AIDC)' and the "*Adaptive Slotine-Li Robot Controller (ADSLRC)*". Additionally, an even more simplified technique is presented in the case of the Modified Adaptive Inverse Dynamics Robot Controller combined with the Sigmoid Generated Fixed Point Transformation.

- The goal of Chapter 3 is to develop a systematic method for the generation of a new family of the *Fixed Point Transformations (FPT)* for the purposes of adaptive control for nonlinear systems. At first the idea is outlined for the *Single Input - Single Output (SISO)* systems. After, it is extended to physical systems having a special $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $n \in \mathbb{N}$ *Multiple Input – Multiple Output (MIMO)* response function. Then, the Thesis makes an attempt to replace the tuning method by a simple calculation.

- Chapter 4 proposes new advances regarding "*Sigmoid Generated Fixed Point Transformation (SGFPT)*". Also, a new control strategy is described based on the combination of the "adaptive" and "optimal" control by applying time-sharing strategy in the SGFPT method, that supports error containment by cyclic control of the different variables. Further, I focus on new improvements on SGFPT technique by introducing "Stretched Sigmoid Functions". The efficiency of the presented control solution is confirmed by the adaptive control of an underactuated mechanical system. Afterwards, I investigate the applicability of fuzzy approximation in the SGFPT-type control design. Additionally, a new type of function is shown for the SGFPT.

- The other important issue that includes the maintenance of unwanted sensor noises that are mainly introduced by feedback into the system under control is discussed in Chapter 5. In the development of a control system the signals of noisy measurements have to be addressed first thus more sophisticated signal pre-processing methods are required. Since, in this Chapter, I focus on the issue of well-adapted techniques for smoothing problems in the time domain and fitting data to parametric models. Widely this means, that research is also needed to determine novel approximations that can well be used for smoothing the operation of the adaptive controller.

- Afterwards, the objective of Chapter 6 is to investigate the Savitzky-Golay (SG) smoothing and differentiation filter. It has been proven that the performance of the classical SG-filter depends on the appropriate setting of the windowlength and the polynomial

degree. Therefore, the main limitations of the performance of this filter are the most conspicuous in processing of signals with high rate of change. Since, in order to evade these deficiencies my aim is to propose a new adaptive design to smooth signals based on the Savitzky-Golay algorithm. The provided method ensures high precision noise removal by iterative multi-round smoothing. The signal approximated by linear regression lines and corrections are made in each step. Also, in each round the parameters are dynamically changed due to the results of the previous smoothing. For supporting high precision reconstruction I introduce a new parametric weighting function.

Finally, the new scientific results are concluded at the end of each Chapter.

1.3 Research Methodology

The theoretical considerations and their usability are validated by simulation investigations. The great majority of the practical problems results in differential equations that do not have solutions in closed analytical form. Since, in order to build numerical simulations I have applied the INRIA's Scilab programming environment. For obtaining realistic simulations I have also applied the SCILAB's XCOS tool that provides an excellent graphical interface and includes more efficient numerical integrators. Furthermore, a few of the simulations have been carried out by using the package "Julia" with a sequential code using Euler integration method. This dynamic language ensures a very fast evaluation for technical computing. For some investigations I have applied Matlab8 that offers a variety of tools and functions that otherwise are widely used in applied research. The applied scientific methods are ensuring the precision and thoroughness of the simulation results.

Chapter 2

Combination of Classical Model Identification with the RFPT-based Design by the Use of a New Tuning Method

The most popular and well studied adaptive control methods in the field of robotics as the "*Adaptive Inverse Dynamics Robot Controller (AIDRC)*" or the "*Adaptive Slotine-Li Robot Controller (ADSLRC)*" apply Lyapunov's 2^{nd} method for tuning the parameters of the actual model of the mechanical system in consideration. The Lyapunov-based technique makes it possible to guarantee the stability of the controlled system using only simple estimations without having any detailed knowledge on its motion that is a great advantage. However, in the application of this technique the main problem is the proper construction of the Lyapunov function. In order to overcome this limitation a possible solution for replacing the Lyapunov technique with the RFPT-based design in these classical controllers firstly raised in [8]. Both the above mentioned classical controllers, namely the AIDRC and ADSLRC were critically analyzed in [8]. It has been shown, that these classical methods can be improved and combined with the RFPT technique. In [8] two modifications were introduced; firstly the parameter tuning processes were modified on the basis of simple geometric interpretation, in order to evade the application of the Lyapunov function in the design: it was shown that consistent tuning of the part of the parameters on which satisfactory information were available was possible without the use of any Lyapunov function. Secondly the feedback term was modified by inserting RFPT-based component, because this modification did not concern the possibilities for parameter tuning. Due to this latter modification the trajectory tracking became precise even in the initial phase of the tuning process in which

the actual parameter estimations were very imprecise.
The essence of parameter tuning is the utilization of the available, geometrically interpreted information, that can be formulated as follows: there is given a *known term* $a \in \mathbb{R}^n$ (it is known partly by measurements and partly by the use of the actual approximate model parameters), a *known matrix* $Y \in \mathbb{R}^{n \times m}$ determined by the precisely modeled kinematic structure of the robot arm, and an *unknown parameter estimation error array* $b \in \mathbb{R}^m$ in the form $a = Yb$, in which $n \in \mathbb{N}$ denotes the degree of freedom of the controlled system, and $m \in \mathbb{N}$ denotes the dimension of the array of the dynamic parameters. The basic idea was to obtain some information on array b. In connection with that, it has to be noted that normally $n \ll m$. In the technical literature for such purposes some *pseudo-inverse* or *generalized inverse* can be used. However, one must be very cautious in choosing an appropriate "inverse":

- In general the solution of this problem is *ambiguous* to the tune of an arbitrary vector $z \neq 0$ for which $Yz = 0$, that is an arbitrary element of the *Null Space* of Y can be added to the solution b: the vector $b + z$ also is the solution of the original problem.

- The elements of this null space also have twofold geometric interpretaion:

  a) An element of this null space corresponds to a non-zero linear combination of the *linearly dependent columns* of matrix Y;

  b) According to the *scalar product of real vectors* the elements of this null-space belong to the *orthogonal subspace of the linear space spanned by the rows* of Y.

- The classical Moore-Penrose pseudoinverse [35, 36] that successfully can be used for solving the inverse kinematic tasks for *redundant robots in the kinematically not singular points* so "distributes" the solution over the available variables that it minimizes the sum $\sum_s b_s^2$. The result is $b = Y^T \left(YY^T\right)^{-1} a$ that is provided as the *linear combination of the rows of* matrix Y. In principle it corresponds to our needs because it cannot contain any element of the null space of the *rows* of Y for which no information is conveyed by the equation under consideration. However, it numerically inconveniently behaves in the vicinity of the singularities where YY^T is ill-conditioned, and it does not exist in the singularities in which $\left(YY^T\right)^{-1}$ cannot be calculated. This problem normally is treated by introducing a small scalar $0 < \mu$ and using an *approximate solution of the original problem*, i.e. $b \approx Y^T \left(YY^T + \mu I\right)^{-1} a$, in which I denotes the *identity matrix of appropriate sizes* (see, [37]). This approximation distorts the existing precise solutions in the non-singular points, and the significance of this distortion can be reduced only by decreasing μ. However, too small μ may result in the

appearance of too big components in the approximate solution. It is evident that the singularities correspond to the elements of the null space of matrix Y^T.

- In order to deal better with the singularities in [38, 39] the application of the *Singular Value Decomposition (SVD)* (for instance [40]) was suggested for the matrix Y in the form: $Y = U\sigma V^T = \sum_l \sigma_l u^{(l)} v^{(l)^T}$ in which $\sigma_l \geq 0$ are the *singular values* of matrix Y, and $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{m\times m}$ are *real orthogonal matrices* the columns of which serve as a *set of orthonormal basis vectors* in $\mathbb{R}^n$ ($\{u^{(l)}\}$) and $\mathbb{R}^m$ ($\{v^{(l)}\}$), respectively. By the use of this basis $b = \sum_l \tilde{b}_l v^{(l)}$, in which (due to the orthonormality of the set) $\tilde{b}_k = v^{(k)^T} b$, and $a = Yb = \sum_l \sigma_l u^{(l)} v^{(l)^T} b = \sum_l \sigma_l u^{(l)} \tilde{b}_l$ can be written. This sum can be reduced only to the positive singular values. Again, due to the orthonormality of the set $\{u^{(l)}\}$, it is obtained that $\tilde{b}_k = \frac{u^{(k)^T} a}{\sigma_k}$, that is $b = \sum_k \frac{u^{(k)^T} a}{\sigma_k} v^{(k)}$. If certain singular values are very small in comparison with the others, we are very uncertain regarding the information content of the original equation in these "directions", so it is expedient to use only the "sure" directions in $b \approx \sum_{k:\sigma_k>\sigma_0} \frac{u^{(k)^T} a}{\sigma_k} v^{(k)}$ in which $\sigma_0 > 0$ is some "limit parameter". Though this approach geometrically can be very well interpreted, its computational need is too high, since both the singular values and the two orthonormal matrices have to be determined for its use.

Based on the above considerations in this chapter my first aim is to investigate the use of the *Modified Gram-Schmidt Algoritm* for the possible combination of the RFPT-based method with a modification of the AIDRC. The algorithm makes the decomposition $Y = \tilde{Y}\Delta$ in which the columns of $\tilde{Y}$ were pairwisely orthogonal but they were not normalized (the original algorithm also executes the normalization of its columns), and Δ denotes an upper triangular matrix with ones in its main diagonals (that was the consequence of omitting the normalizations). The presented approach referred to took it into consideration that in a given control step we do not need the solution to an *arbitrary array a_{arb} in the LHS* of $a_{arb} = Yb$: we need the solution only for a given array a. Since we did not need a *complete generalized inverse*, the computation needs were reduced in calculating or estimating b. The problem of the "uncertain directions" was treated in a similar way as in the case of the SVD-based solution: in $\tilde{Y}$ in the place of the linearly dependent columns zeros appear, and wery small contributions are present for those directions for which little independent components remains. These columns can be replaced by zeros in $\tilde{Y}$, and the approximation of b can be built up by the use of this modified $\tilde{Y}_{approx}$. Due to their structures the inverse matrices of $\tilde{Y}$ and Δ can be built up in a relatively easy way, that is detailed in the following sections. Following that, I show a same possibility for the *Modified Adaptive Slotine-Li Robot Controller (MADSLRC)*. Finally a new, even simpler tuning technique is presented for the Modified Adaptive Inverese Dynamic Robot Controller.

2.1 Principles of the Original Robust Fixed Point Transformation for Nonlinear Control

As an alternative of the Lyapunov function technique in adaptive control the method of *"Robust Fixed Point Transformations" (RFPT)* was suggested in [41]. This approach assumes the existence of an *approximate dynamic model* used by the controller for the calculation of the control "forces" belonging to some purely kinematically prescribed trajectory tracking error reduction (it is the "desired response" of the system, r^{Des}), and compares it with the *actually observed response* r^{Act}, that is formed according to the *exact dynamics* of the system under control. In this manner a "response function" $r^{Act} = f(r^{Des}, \ldots)$ can be introduced, in which normally $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $n \in \mathbb{N}$ for a MIMO system. In the argument list of f the symbol "$\ldots$" represents the state variables and the unknown environmental "forces" that also influence the system's response. (Depending on the phenomenology of the controlled system the responses may be some –generally higher order– time-derivatives of the system's coordinates, while the "forces" may mean force or torque values for mechanical systems, voltages or currents for electrical ones, or the input rates of some reagents in the case of chemical reactions, etc.) Due to the modeling errors and the unknown external disturbances, normally $r^{Act} \neq r^{Des}$. The basic idea was an application of Stefan Banach's *Fixed Point Theorem* [42] in the following manner: instead of tuning the model parameters or the feedback gains for the calculation of the control "forces", the controller generates an iterative sequence of the *"Deformed Responses"* $\{r_n\}$ that are introduced into the approximate dynamic model instead of r^{Des}. If this sequence converges to a *"deformed input value"* $r_\star$ so that $r^{Des} = f(r_\star, \ldots)$, the input of the approximate model is appropriately deformed. To obtain a sequence that converges to the solution of the control task an iteration was generated by a *contractive map* over a complete linear metric space. The abbreviation "RFPT" refers to a nonlinear map that, in combination with the response function, generates the convergent sequences. It was shown that on the basis of the same idea MRAC controllers can be easily designed without extra mathematical considerations [17]. The idea was also extended to MIMO systems. In comparison with the Lyapunov function based technique, the RFPT has the features as follows: *a)* the method is very simple and easily implementable; *b)* it keeps in the center of attention the primary design intent, i.e. the kinematically formulated tracking error relaxation; *c)* it works only with a few adaptive parameters that are clearly set; *d)* it does not impose unnecessary conditions to be met; *d)* its weak point is that *in its basic form* cannot guarantee global stability. The basin of convergence of the sequence is bounded and theoretically it may happen that the control signals leave this basin. Recent investigations revealed that in this case the control signal may produce chattering [26, 27]. It

was also shown that by tuning one of its altogether 3 adaptive parameters, for a wide class of physical systems, the controller can be kept within the basin of attraction [24] and that by the use of model-independent observers, "precursor oscillations" can be observed in the control signal that are not dangerous for the control since they belong to non-monotonic, oscillating convergence to the solution of the control task [43].

2.2 Critical Analysis and Modification of the AIDC Controller

The method's main properties are demonstrated by numerical simulations regarding the control of a 1 DoF paradigm, the 5^{th} order modification of the van der Pol oscillator, a nonlinear physical system that produces nonlinear oscillations first analyzed, modeled, and understood by van der Pol in 1927 [44]. This system has the equation of motion as

$$m\ddot{q} + \mu\left(q^2 - c\right)\dot{q} + kq + \beta q^3 + \lambda q^5 = F \tag{2.1}$$

in which $m = 10$ physically corresponds to some *inertia*, $\mu = 1$ describes some viscous damping if $q^2 > c$ otherwise it means energy input, $c = 3$ determines the limit between the damped and excited regions, $k = 100$ corresponds to the stiffness of a linear spring while $\beta = 1$ and $\lambda = 2$ mean nonlinear corrections in the third and fifth order, that is the stiffness of the spring drastically increases with its dilatation, q. Variable F describes the external forces acting on this system. (Since my investigations are of mathematical nature, for the sake of simplicity the physical dimensions/units of the various quantities will not be considered in this example).

2.2.1 The Operation of the Classical AIDC Controller

It can be observed that (2.1) satisfies the conditions that must be met to construct an AIDC according to [45]: *the dynamic parameters of the system model* can be linearly separated into an array that is multiplied by known or measurable functions of the state variables of the system q, $\dot{q}$, and $\ddot{q}$ in the form $F = Y\left(\ddot{q}, \dot{q}, q\right)\Theta$ as

$$\begin{aligned} Y &= \left(\ddot{q}, \dot{q}q^2, -\dot{q}, q, q^3, q^5\right), \\ \Theta &= \left(m, \mu, \mu c, k, \beta, \lambda\right)^T. \end{aligned} \tag{2.2}$$

Let the *available approximate model parameters* be $\hat{m} = 9$, $\hat{\mu} = 2$, $\hat{c} = 3.5$, $\hat{k} = 110$, $\hat{\beta} = 0.9$, and $\hat{\lambda} = 1.5$.

Assuming that for the *nominal trajectory* the $q^N(t)$, $\dot{q}^N(t)$, $\ddot{q}^N(t)$ values are known by the use of a kinematic PD-type feedback described by the gains K_1 and K_2, this controller applies the modified $\ddot{q}^N + K_1\left(q^N - q\right) + K_2\left(\dot{q}^N - \dot{q}\right)$ acceleration instead of $\ddot{q}^N$, and applies the approximate model parameters for the estimation of the necessary force as

$$\begin{gathered} F = \hat{m}\left[\ddot{q}^N + K_1\left(q^N - q\right) + K_2\left(\dot{q}^N - \dot{q}\right)\right] + \\ \hat{\mu}\left(q^2 - \hat{c}\right)\dot{q} + \hat{k}q + \hat{\beta}q^3 + \hat{\lambda}q^5 \end{gathered} \tag{2.3}$$

that – in the lack of external perturbations – must be identical to the force in (2.1) containing the *exact parameters*. The control force F evidently can be eliminated from (2.1) and (2.3). Furthermore, it is easy to see that by subtracting $\hat{m}\ddot{q}$ from both sides of the so obtained equation the appropriate time-derivatives of the tracking error $e(t) \stackrel{def}{=} q^N(t) - q(t)$ appears in the left-hand side (LHS) as

$$\begin{gathered} \hat{m}\left[\ddot{e} + K_1 e + K_2 \dot{e}\right] + \hat{\mu}\left(q^2 - \hat{c}\right)\dot{q} + \\ \hat{k}q + \hat{\beta}q^3 + \hat{\lambda}q^5 = \\ (m - \hat{m})\ddot{q} + \mu\left(q^2 - c\right)\dot{q} + kq + \beta q^3 + \lambda q^5. \end{gathered} \tag{2.4}$$

In the next step it can be observed that by rearranging (2.4) in the RHS the *modeling error* appears as

$$\hat{m}\left[\ddot{e} + K_1 e + K_2 \dot{e}\right] = Y\left(\ddot{q}, \dot{q}, q\right)\left(\Theta - \hat{\Theta}\right). \tag{2.5}$$

In order to find a Lyapunov function in the design of the AIDC controller the variable $x \stackrel{def}{=} (e, \dot{e})^T$ "*artificial state variable*" is introduced and the equation of motion can be rewritten as

$$\begin{gathered} \dot{x} = \begin{pmatrix} 0 & 1 \\ -K_1 & -K_2 \end{pmatrix} x + \begin{pmatrix} 0 \\ \frac{Y}{\hat{m}} \end{pmatrix}\left(\Theta - \hat{\Theta}\right) \stackrel{def}{=} \\ Ax + B. \end{gathered} \tag{2.6}$$

With the symmetric positive definite matrices P (of size 2×2) and R (of size 6×6) a Lyapunov function is defined as

$$V \stackrel{def}{=} x^T P x + \left(\Theta - \hat{\Theta}\right)^T R \left(\Theta - \hat{\Theta}\right). \tag{2.7}$$

Equation (2.7) evidently defines a good Lyapunov function that takes zero if and only if the errors x and $\left(\Theta - \hat{\Theta}\right)$ equal to zeros. According to the "orthodox" solution $\dot{V}$ must be made negative [A. 3]:

$$\dot{V} = \dot{x}^T P x + x^T P \dot{x} + 2\left(\dot{\Theta} - \dot{\hat{\Theta}}\right)^T R \left(\Theta - \hat{\Theta}\right) < 0 \tag{2.8}$$

in which the symmetry of R is utilized. Via substituting (2.6) into (2.8) the term quadratic in x can be separated and by using the symmetry of P the remaining terms linear in x can be collected as

$$\dot{V} = x^T \left(A^T P + PA\right) x + \\ 2\left\{\left(\dot{\Theta} - \dot{\hat{\Theta}}\right)^T R + x^T P \begin{pmatrix} 0 \\ \frac{Y}{\hat{m}} \end{pmatrix}\right\}\left(\Theta - \hat{\Theta}\right). \tag{2.9}$$

For making $\dot{V}$ negative the quadratic part can be made negative by solving the *Lyapunov equation* for a positive definite matrix Q as $\left(A^T P + PA\right) = -Q$ and the remaining part can be made zero by the parameter tuning rule (2.10) since $\dot{\Theta} \equiv 0$, where

$$\dot{\hat{\Theta}}^T = x^T P \begin{pmatrix} 0 \\ \frac{Y}{\hat{m}} \end{pmatrix} R^{-1}. \tag{2.10}$$

The Lyapunov equation has appropriate solution if the real part of the eigenvalues of A are negative. For this the fedback gains K_1 and K_2 must be properly chosen.

It is worth noting that the name of the method follows from the fact that in (2.10) the *inverse of the estimated inertia* $\hat{m}$ occurs that considerably limits the speed of parameter tuning. If the numerical algorithm achieves the $1/0$ singularity the learning process stops without useful results. Another weak point of the solution is that the effects of the unknown external disturbances are improperly compensated by the tuning process [A. 3]. Furthermore P and R contain numerous arbitrary parameters. To exemplify the operation of the method for $\Lambda = 10$, $K_1 = \Lambda^2$, $K_2 = 2\Lambda$, $Q = \langle 100, 100 \rangle$, $R = \langle 5, 5, 5, 5, 5, 5 \rangle$ simulation results are shown in Figs. 2.1–2.4.

It can be seen that the learning speed of the controller is very small. It was found that in the case of some decrease in R caused singularities in the tuning process. To improve the situation similar steps are done as in [8].

2.2.2 Modified Tuning Algorithm

In order to speed up the learning process it would be expedient to avoid the use of $\hat{m}^{-1}$ in the tuning algorithm [A. 1]. This program seems to be possible if we return to (2.5) and note that the LHS of the equation is known and Y is also known in the right-hand side (RHS). This means that (2.5) provides *actual and available information* on the projection of the modeling error $\Theta - \hat{\Theta}$ in the direction of Y^T. If we do not want to use any Lyapunov function for parameter tuning, this information can be utilized directly by the tuning rule (2.11) that can be expounded as follows: an exponential decay with the exponent $-\alpha$ could be resulted by the equation $\frac{\mathrm{d}(\Theta - \hat{\Theta})}{\mathrm{d}t} = -\alpha(\Theta - \hat{\Theta})$. However, instead of the full parameter

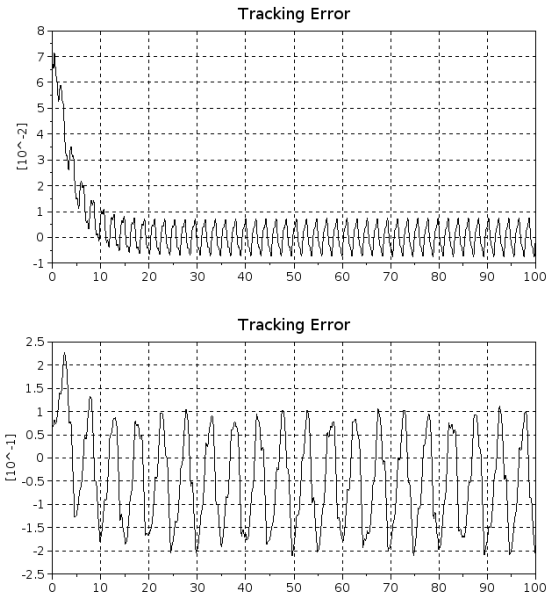

Figure 2.1: The trajectory tracking error of the AIDC in the case free of external disturbances (upper chart) and in the case of disturbance forces (3^{rd} order spline functions of time) (lower chart)

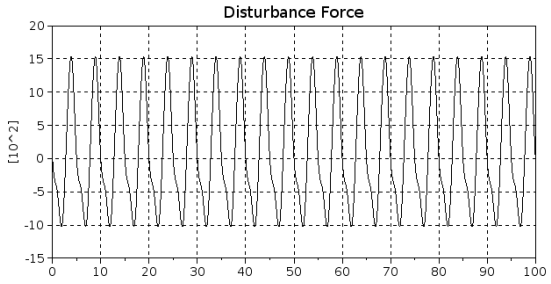

Figure 2.2: The disturbance forces pertaining to the lower chart of Fig. 2.1

error vector only its projection to the direction of Y^T is known that can be generated by a projector as $\frac{Y^T Y}{\|Y\|^2}(\Theta - \hat{\Theta})$. According to (2.5) $Y(\Theta - \hat{\Theta}) = \hat{m}\,(\ddot{e} + K_1 e + K_2 \dot{e})$. Therefore if *exponential decay of the known components* is required only, then (2.11) can be deduced (ε stands to avoid division by zero).

$$\dot{\hat{\Theta}} = \alpha \hat{m}\,(\ddot{e} + K_1 e + K_2 \dot{e})\,\frac{Y^T}{\|Y\|^2 + \varepsilon} \tag{2.11}$$

To show the applicability of this new tuning for $\alpha = 10$ simulation results are given in Figs. 2.5–2.7. The exact parameters are $\Theta = (10, 1, 3, 100, 1, 2)^T$. It is evident that Θ_2, Θ_3, Θ_5 and Θ_6 are well learned but Θ_1 and Θ_4 are only slowly approximated. In the case of the new tuning it cannot taken for granted that the exact value of each system parameter

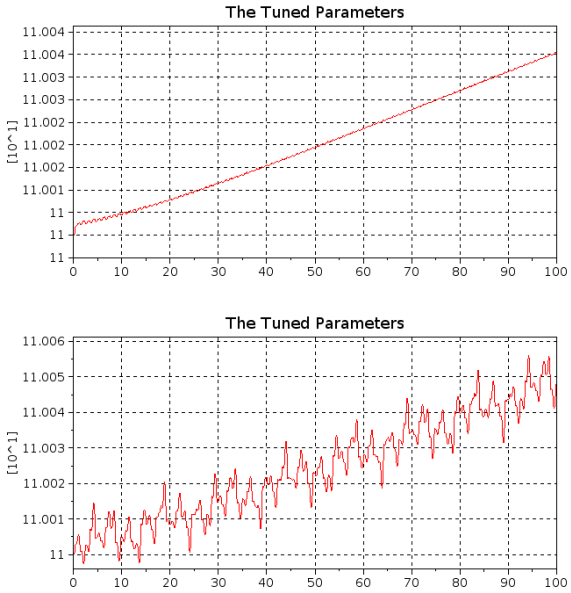

Figure 2.3: Tuning of parameter $\Theta_4 \equiv \hat{k}$ without (upper chart) and with (lower chart) external disturbances

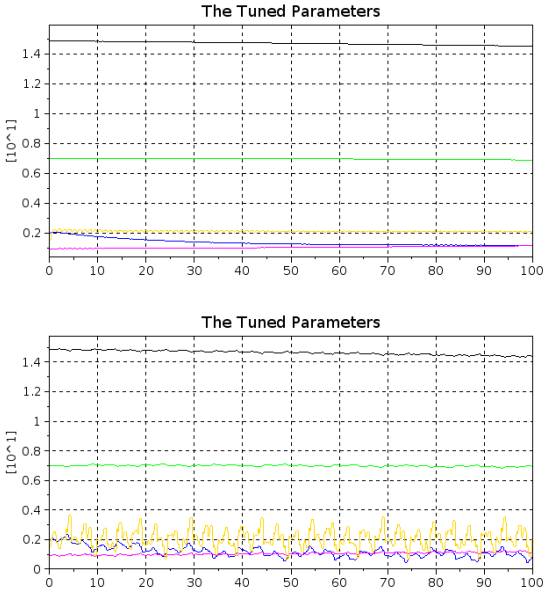

Figure 2.4: Tuning the other parameters in Θ without (upper chart) and with (lower chart) external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines]

will be precisely learned: if the occurring motion does not yield satisfactory information on certain parameters these parameters will not be learned. However, it also means that the

exact knowledge on these parameters is not necessary for guaranteeing precise tracking.

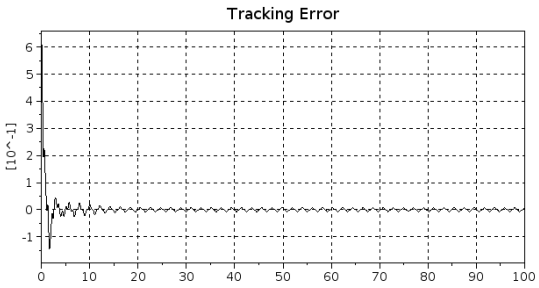

Figure 2.5: The trajectory tracking error of the AIDC with modified tuning in the case free of external disturbances

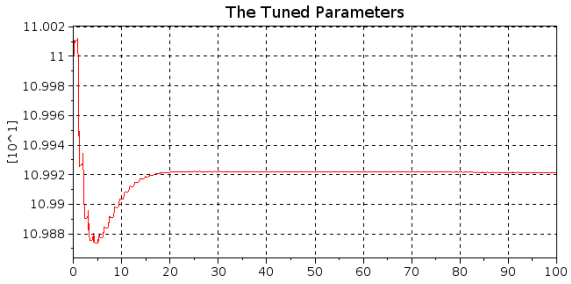

Figure 2.6: Tuning of parameter $\Theta_4 \equiv \hat{k}$ of the AIDC with modified tuning without external disturbances

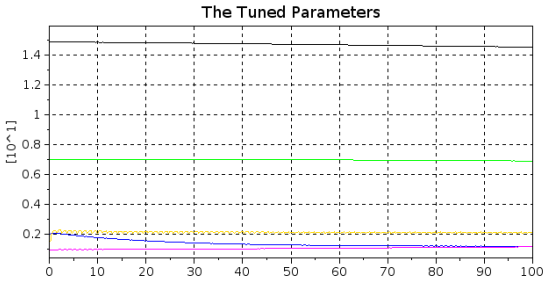

Figure 2.7: Tuning the other parameters in Θ of the AIDC with modified tuning without external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines]

According to the simulations this speedy parameter learning is very vulnerable by even small external disturbances. Even very small ones made the simulation results diverge. For $\alpha = 1$ and a very limited disturbance force results are displayed in Figs. 2.8–2.9. It is evident that the external perturbations also disturb this new tuning method too, and decrease the

quality of trajectory tracking. In the next section it will be shown that the combination of parameter tuning with the RFPT-based adaptive technique can seriously improve the situation [A. 1].

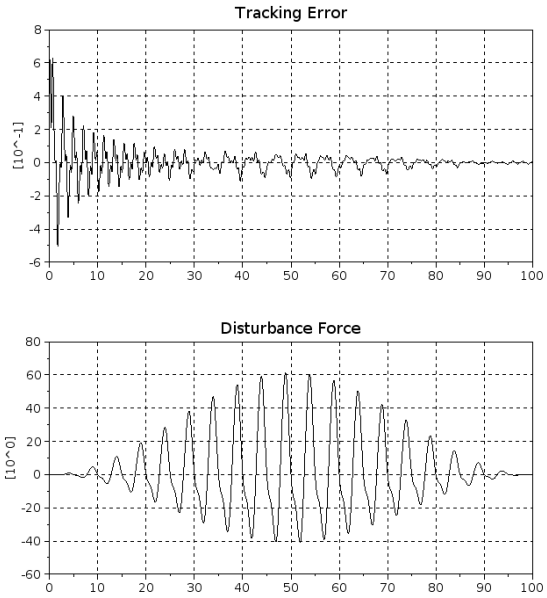

Figure 2.8: The trajectory tracking error of the AIDC (upper chart) with modified tuning and limited external disturbances (lower chart)

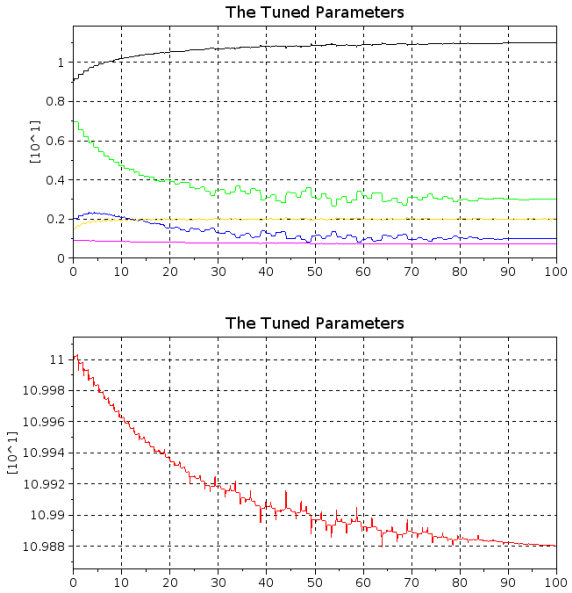

Figure 2.9: Tuning the parameters in Θ of the AIDC with modified tuning with reduced external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_4 \equiv \hat{k}$: red, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines]

2.3 Combination of the New Tuning Method with the RFPT-based Adaptive Control

The idea comes from the observation that in the lack of unknown external perturbations a more general variant of (2.5) can be deduced as (2.12) in which the "*Required Acceleration*" $\ddot{q}^{Req}$ can be freely determined by a *kinematically prescribed trajectory tracking mode*. (If we do not wish to introduce a Lyapunov function for the deduction of parameter tuning, we have a great *formal freedom*.) If exponential error relaxation is prescribed as $\left(\frac{\mathrm{d}}{\mathrm{dt}} + \Lambda\right)^2 \left(q^N(t) - q(t)\right) = 0$, a "*Desired Acceleration*" $\ddot{q}^{Des} \stackrel{def}{=} \ddot{q}^N + \Lambda^2 \left(q^N - q\right) + 2\Lambda(\dot{q}^N - \dot{q})$ can be introduced just as it was done in the above simulations.

$$\hat{m}\left[\ddot{q}^{Req} - \ddot{q}\right] = Y\left(\ddot{q}, \dot{q}, q\right)\left(\Theta - \hat{\Theta}\right). \tag{2.12}$$

Let the control signal be *iteratively determined* for the consecutive control cycles with the sigmoid function $\sigma(x) \stackrel{def}{=} \frac{x}{|x|+1}$ as given in (2.13) in which $\ddot{q}_{n-1}$ is the *observed system response* for the control signal $\ddot{q}^{Req}_{n-1}$.

$$\begin{aligned} \ddot{q}^{Req}_n \stackrel{def}{=} &\left(\ddot{q}^{Req}_{n-1} + K_c\right) \times \\ &\left\{1 + B_c\sigma(A_c[\ddot{q}_{n-1} - \ddot{q}^{Des}_n])\right\} - K_c. \end{aligned} \tag{2.13}$$

This equation evidently is an adaptive structure that learns from the past behavior of the system. The properties of the RFPT defined by (2.13) have been widely studied in [41], [26], [43]. Here we only note that if $\ddot{q}_{n-1} = \ddot{q}_n^{Des}$ then $q_n^{Req} = q_{n-1}^{Req}$, that is the solution of the control task is the *fixed point* of this mapping. The behavior of this controller as well as the simple methods for setting its adaptive control parameters have been widely investigated and it was found that for a wide class of physical systems it simultaneously and efficiently can compensate the effects of unknown external disturbances and modeling errors. *Therefore it is expected that by the use of its control signals in the lack of external perturbations the AIDC with modified tuning can learn the system parameters, and in the case of external perturbations it can compensate their effects even if the parameter tuning process becomes improper* [A. 1]. In the sequel simulation results will be provided that substantiate this statement.

2.3.1 Simulation Results for the RFPT-supported AIDC Controller with Modified Tuning Rule

The simulations were made for the control parameter settings $K_c = -10^5$, $B_c = 1$, and $A = 10^{-6}$. Figure 2.10 shows the tracking error and the disturbance forces. In Fig. 2.11 it can be seen that the external perturbations again "mislead" the parameter tuning process. Figure 2.12 reveals that the phase trajectory tracking remained nice and precise in spite of the considerable external disturbances.

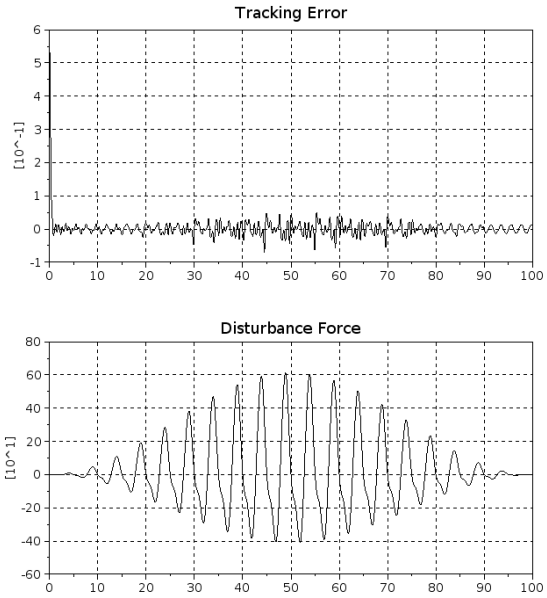

Figure 2.10: The trajectory tracking error of the RFPT-supported AIDC (upper chart) with modified tuning and considerable external disturbances (lower chart)

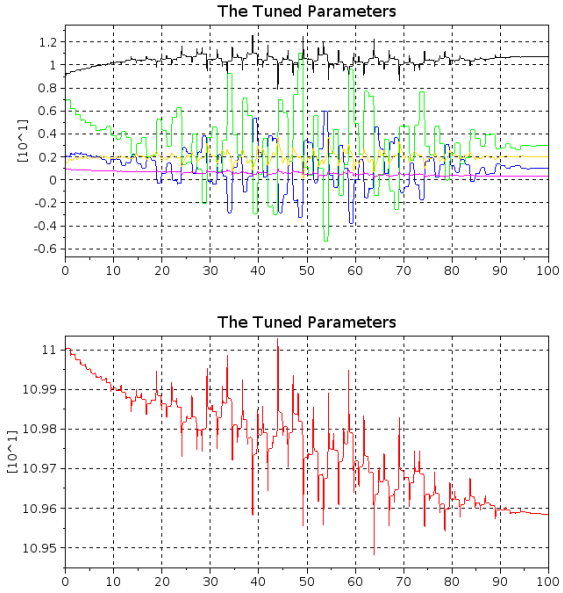

Figure 2.11: Tuning the parameters in Θ of the RFPT-supported AIDC with modified tuning with considerable external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_4 \equiv \hat{k}$: red, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines]

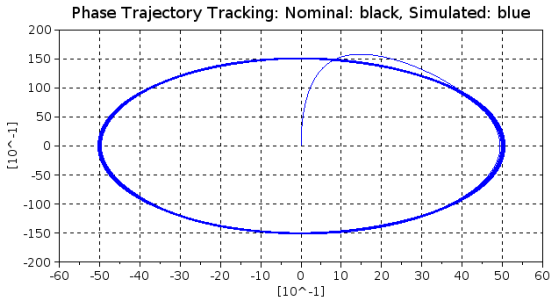

Figure 2.12: The phase trajectory tracking of the RFPT-supported AIDC with modified tuning and considerable external disturbances

2.4 Combination of the Modified Adaptive Slotine-Li Robot Controller (ADSLRC) with the RFPT-based Adaptive Controller

In the present section my aim is to show a similar possibility for the ADSLRC controller [A. 2]. For starting point we go back to its modification introduced in [8]. For simulation purposes and illustrations the same paradigm (a cart+beam+hamper system) will be used here.

2.4.1 The Tuning Method using Lyapunov-function

The *integrated tracking error* can be introduced as $\xi(t) \stackrel{def}{=} \int_{t_0}^{t} \left[q^N(\zeta) - q(\zeta)\right] \mathrm{d}\zeta$. If $\Lambda > 0$ (constant symmetric positive definite matrix) an "*error metrics*" can be introduced as $S(t) \stackrel{def}{=} \left(\frac{\mathrm{d}}{\mathrm{d}t} + \Lambda\right)^2 \xi(t)$. Furthermore, for the feedback the quantity $v \stackrel{def}{=} \dot{q}^N + 2\Lambda\dot{\xi} + \Lambda^2\xi$ also is practically defined. Evidently $v - \dot{q} = S$.

As it was shown by Slotine and Li, the *approximate model* of the robot can be described by the *positive definite symmetric inertia matrix* $\hat{H}(q)$, the *special matrix* $\hat{C}(q, \dot{q})$, the approximation of the gravitational term $\hat{g}(q)$, and a positive symmetric matrix K_D, in which variable q denotes the "*Generalized coordinates*" of the robot. Regarding the definition of matrix C this method takes into account the fact that in the Euler-Lagrange equations of motion this matrix is composed from the inertia matrix:

$$\begin{gathered} L \stackrel{def}{=} \frac{1}{2}\sum_{ij} H_{ij}\dot{q}_i\dot{q}_j - U(q), \\ Q_k = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k}, \\ Q_k = \sum_j H_{kj}\ddot{q}_j + \sum_{ji} \frac{\partial H_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j \\ -\frac{1}{2}\sum_{ij} \frac{\partial H_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j + \frac{\partial U}{\partial q_k}, \end{gathered} \tag{2.14}$$

in which the product $\dot{q}_i\dot{q}_j$ is *symmetric* in the indices i, j, therefore only the symmetric part of its coefficient yields contribution as

$$\begin{gathered} Q_k = \sum_j H_{kj}\ddot{q}_j + \frac{\partial U}{\partial q_k} + \\ \sum_{ji} \left(\frac{1}{2}\frac{\partial H_{kj}}{\partial q_i} + \frac{1}{2}\frac{\partial H_{ki}}{\partial q_j} - \frac{1}{2}\frac{\partial H_{ij}}{\partial q_k}\right)\dot{q}_i\dot{q}_j \\ C_{kj} \stackrel{def}{=} \frac{1}{2}\sum_i \left(\frac{\partial H_{kj}}{\partial q_i} + \frac{\partial H_{ki}}{\partial q_j} - \frac{\partial H_{ij}}{\partial q_k}\right)\dot{q}_i \end{gathered} \tag{2.15}$$

Let the controller exert the generalized force Q according to (2.16). An important assumption of the method is that neither unknown external disturbances nor other modeling inaccuracies may exist, therefore the generalized force Q as calculated in the first line of (2.16) is related to the motion of the system as given by its 2^{nd} line:

$$
\begin{aligned}
Q = \hat{H}(q)\dot{v} + \hat{C}(q,\dot{q})v + \hat{g}(q) + K_D S \\
Q = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \\
= Y(q,\dot{q},v,\dot{v})\Theta,
\end{aligned}
\tag{2.16}
$$

in which the "exact model values" are denoted by $H(q)$, $C(q,\dot{q})$, and $g(q)$, and it is also utilized that the array of the dynamic model parameters Θ can be written in a separated form in which Y is *exactly known*.

The equality of the left hand sides of the equations in (2.16) traditionally is utilized as follows. Following the elimination of Q from both sides the *unknown quantities* (the exact matrices are not known) $H\dot{v}$, Cv, g, and $K_D S$ can be subtracted. Since $-H\dot{v} + \ddot{q} = -H\dot{S}$, and $C(-v + \dot{q}) = -CS$, it is obtained that

$$
\begin{aligned}
(\hat{H} - H)\dot{v} + (\hat{C} - C)v + (\hat{g} - g) = \\
-H\dot{S} - CS - K_D S = Y(\hat{\Theta} - \Theta).
\end{aligned}
\tag{2.17}
$$

The Lyapunov function is $V = \frac{1}{2}S^T H(q)S + \frac{1}{2}(\Theta - \hat{\Theta})^T \Gamma (\Theta - \hat{\Theta})$. For guaranteeing negative time-derivative for the Lyapunov function

$$
\begin{aligned}
\dot{V} = S^T H\dot{S} + \tfrac{1}{2}S^T \dot{H} S + \\
(\dot{\Theta} - \dot{\hat{\Theta}})^T \Gamma (\Theta - \hat{\Theta})
\end{aligned}
\tag{2.18}
$$

must be made negative. From (2.17) $H\dot{S}$ can be expressed and substituted into (2.18):

$$
\begin{aligned}
\dot{V} = S^T \left(-Y(\hat{\Theta} - \Theta) - CS - K_D S \right) + \\
S^T \tfrac{1}{2}\dot{H} S + (\dot{\Theta} - \dot{\hat{\Theta}})^T \Gamma (\Theta - \hat{\Theta}).
\end{aligned}
\tag{2.19}
$$

Taking into account that according to (2.15) $\frac{1}{2}\dot{H}_{kj} - C_{kj} = \frac{1}{2}\sum_i \left(-\frac{\partial H_{ki}}{\partial q_j} + \frac{\partial H_{ij}}{\partial q_k} \right) \dot{q}_i$ is *skew symmetric* in the indices (k,j), $S^T \left(\frac{1}{2}\dot{H} - C \right) S = 0$, and the the condition of the stability is

$$
\begin{aligned}
0 > \dot{V} = -S^T K_D S + \\
\left[S^T Y + (\dot{\Theta} - \dot{\hat{\Theta}})^T \Gamma \right] (\Theta - \hat{\Theta}).
\end{aligned}
\tag{2.20}
$$

Since normally $\dot{\Theta} \equiv 0$ and K_D is positive definite the appropriate parameter tuning rule can be: $\dot{\hat{\Theta}}^T = S^T Y \Gamma^{-1}$. It is worths noting that:

- since in this approach no matrix inversion happens, the speed of parameter tuning can be quite high;
- the actual value of $\dot{V}$ is *independent of* $(\Theta - \hat{\Theta})$ and $\frac{\mathrm{d}}{\mathrm{d}t}(\Theta - \hat{\Theta})$, therefore if the $S = 0$ state is achieved, *the parameter tuning process is stopped even if the estimation error is*

not zero, and the consequence of any instant disturbance that kicks out S from zero is an immediate decrease in $\|S\|$;

- this method cannot properly compensate the effects of unknown external disturbances and friction forces since in the first two lines of (2.16) the same Q generalized force must occur;
- further problems arise with the systems for which the model cannot be separated as a multiplication of the array of the dynamical parameters and known functions [A. 2].

The above statements are trivial and do not require illustration via simulation. In the next subsection it will be shown that consistent parameter tuning can be invented without the use of any Lyapunov function.

2.4.2 New Parameter Tuning

Let us return to (2.16) and observe that if the aim is not the construction of any Lyapunov function, the *known terms* as $\hat{H}\ddot{q}$, $\hat{C}\dot{q}$, and $\hat{g}$ can be subtracted from both sides of the equation that was obtained after the elimination of Q. In the result we again obtain the *modeling error* multiplied by *known quantities* at one side and known quantities will appear at the other side [A. 2]:

$$\begin{gathered}\hat{H}(q)(\dot{v}-\ddot{q})+\hat{C}(q,\dot{q})(v-\dot{q})+K_D S= \\ \left[H-\hat{H}\right]\ddot{q}+\left[C-\hat{C}\right]\dot{q}+[g-\hat{g}]= \\ =Z(q,\dot{q},\ddot{q})\left(\Theta-\hat{\Theta}\right)\end{gathered} \tag{2.21}$$

in which $Z(q,\dot{q},\ddot{q})$ is a *known quantity*. This is a great advantage with respect to (2.17) in which the left hand side of the 2^{nd} equation is not known since H and C are unknown. Equation (2.21) has *simple geometric interpretation that directly can be used for parameter tuning* as follows: if *exponential decay rate* could be realized for the parameter estimation error, the *array equation* $\frac{\mathrm{d}}{\mathrm{d}t}\left(\Theta-\hat{\Theta}\right)=-\alpha\left(\Theta-\hat{\Theta}\right)$ ($\alpha>0$) should be valid. If we multiply both sides of this equation with a *projector* determined by *a few pairwisely orthogonal unit vectors* as $\sum_i e^{(i)}e^{(i)^T}$ the equation $\sum_i\left(\dot{\Theta}_i-\dot{\hat{\Theta}}_i\right)=-\alpha\sum_i e^{(i)}\left(\Theta_i-\hat{\Theta}_i\right)$ is obtained. This situation can well be approximated if we use the Gram-Schmidt algorithm ([46], [47]) for finding the *orthogonal components* of the rows of matrix Z in (2.21). Assuming that the speed of variation of Z is not too significant, we can apply the tuning rule *only for the known components* in the form [A. 2]: $\frac{\mathrm{d}}{\mathrm{d}t}(\Theta-\hat{\Theta})=-\alpha\sum_i\frac{\tilde{z}^{(i)}\tilde{z}^{(i)^T}}{\|\tilde{z}^{(i)}\|^2+\varepsilon}(\Theta-\hat{\Theta})$ in which $\tilde{z}^{(i)}$ denotes the transpose of the orthogonalized rows of matrix Z, and a small $\varepsilon>0$ evades division by zero whenever the norm of the appropriate row is too small. Since the scalar product is a *linear operation* during the orthogonalization process the appropriate linear combinations of the scalar products in the 3^{rd} row of (2.21) can be computed.

2.4.3 Further Modification in the Exerted Force/Torque Components

It is evident that all the above considerations remain valid if in the place of $\hat{H}\dot{v}$ some different term is written in (2.21) [A. 2]. (Obtaining exactly $\dot{S}$ was important only for the construction of a Lyapunov function.) So useful information can be obtained for model parameter tuning if in the exerted forces this term is replaced by its iterative variant obtained from the RFPT-based design as follows:

$$\begin{gathered} h := f(r_n) - r_{n+1}^d, \quad e := h/\|h\|, \\ \tilde{B} = B_c\sigma(A_c\|h\|) \\ r_{n+1} = (1+\tilde{B})r_n + \tilde{B}K_c e \end{gathered} \tag{2.22}$$

in which $\sigma(x) \stackrel{def}{=} \frac{x}{1+|x|}$, $r_{n+1}^d \stackrel{def}{=} v_{n+1}$, r_n denotes the adaptively deformed control signal used instead of v_n control in control cycle n, and $f(r_n) \equiv \ddot{q}_n$, i.e. the *observed system response* in cycle n. It is evident that if $f(r_n) = r_{n+1}^d$ then $r_{n+1} = r_n$, that is the solution of the control task (i.e. the appropriate adaptive deformation) is the fixed point of the mapping defined in (2.22). Since the details of the convergence were discussed in ample literature references in the sequel only simulation results will be presented to reveal the cooperation of the RFPT-based adaptivity and model parameter tuning.

2.4.4 Simulation Results

For the simulations the same cart+beam+hamper system was used as in [48] with the Euler–Lagrange equations of motion

$$\begin{gathered} \begin{bmatrix} (ML^2+\theta) & \theta & mLcosq_1 \\ \theta & \theta & 0 \\ mLcosq_1 & 0 & (m+M) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \\ + \begin{bmatrix} -mgLsinq_1 \\ 0 \\ -mLsinq_1\dot{q}_1^2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}. \end{gathered} \tag{2.23}$$

in which $M = 30\,kg$ and $m = 10\,kg$ denote the masses of the cart and the hamper, respectively (the mass of the beam connecting the hamper to the cart is neglected), $\theta = 20\,kg \cdot m^2$ describes the momentum of the hamper referenced to its rotary axle on which its mass center point is located, $L = 2\,m$ denotes the length of the beam, and $g = 10\,m/s^2$ in this case denotes the gravitational acceleration. With the definition $\Theta \stackrel{def}{=} [mL, mL^2+\theta, \theta, M+m, mgL]^T$ matrix Z easily can be constructed. The *approximate model parameters* are $\hat{M} = 60\,kg$ and $\hat{m} = 20\,kg$, $\hat{\theta} = 50\,kg \cdot m^2$, $\hat{L} = 2.5\,m$ (in the dynamical calculations), and $\hat{g} = 8\,m/s^2$. These settings correspond to $\hat{\Theta}_{ini} = [50, 175, 50, 80, 400]^T$, and

$\Theta = [20, 60, 20, 40, 200]^T$.

2.4.5 Cooperation in the Lack of External Disturbances

In the first step it will be illustrated that the RFPT-based design can well coexist with the dynamical parameter tuning in the absence of disturbances [A. 2]. The control parameters are as follows: $\Lambda = 10/s$, $\alpha = 1/s$, $K_D = 100/s$, $K_c = -10^7$, $B_c = 1$, and $A_c = 10^{-8}$, the cycle time and the time-resolution of the numerical (Euler-type) integration was $\delta t = 10^{-4}\, s$. According to Fig. 2.13 the application of the RFPT considerably improved the tracking precision. As it is displayed by Fig. 2.14 the initially strongly over-estimated parameters are tuned in similar manner.

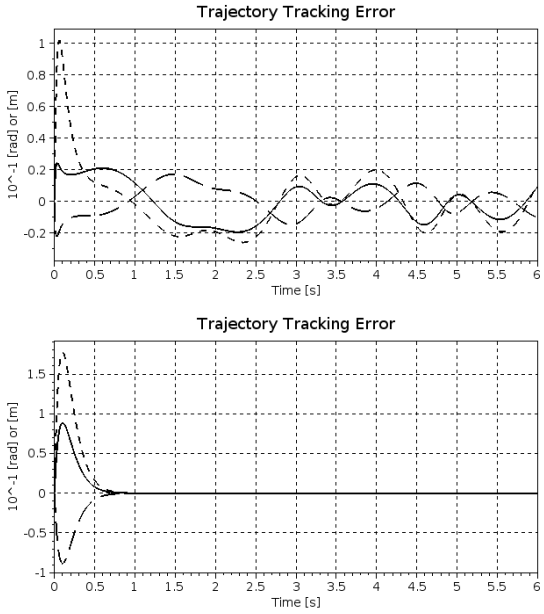

Figure 2.13: The tracking error in the lack of unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines]

2.4.6 Cooperation under the Effect of a LuGre Friction at Axle 3

For disturbances a LuGre-type (Lund-Grenoble) friction was introduced at axle 3 as it was done in [48]. This model cannot be taken into account in a "separated form" and also contains an internal dynamic variable that is not modeled by our controller (it is used only in the simulations). Figure 2.15 reveals that the application of the RFPT again considerably improves the tracking error, with the exception of the initial "transient" section.

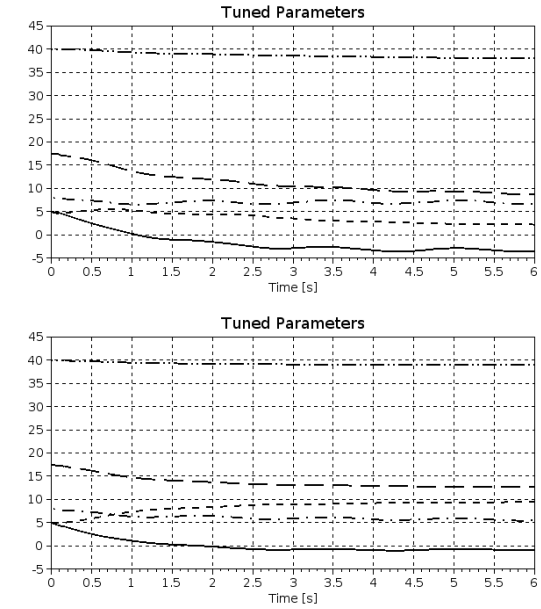

Figure 2.14: Tuning of the adaptive parameters in the lack of unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[Θ_1: solid, Θ_2: dashed, Θ_3: dense dash, Θ_4: dash-dot, and Θ_5: dash-dot-dot lines]

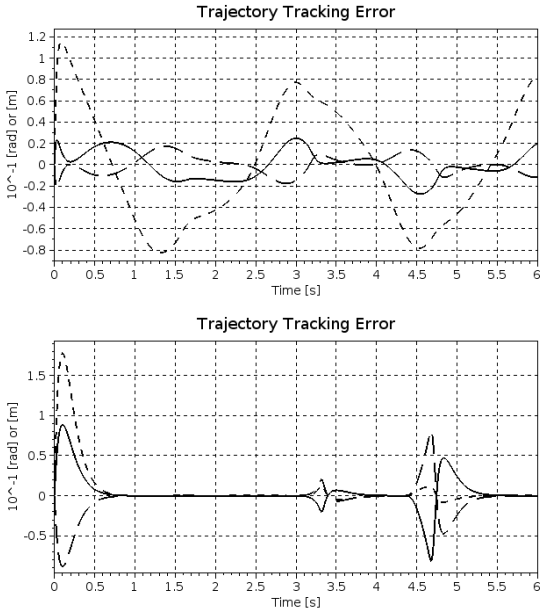

Figure 2.15: The tracking error under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines]

Figure 2.16 reveals that similar "abnormal" tuning-discrepancies occur in both cases, but the

RFPT-based method well compensates the simultaneous consequences of the disturbances and improper parameter tuning.

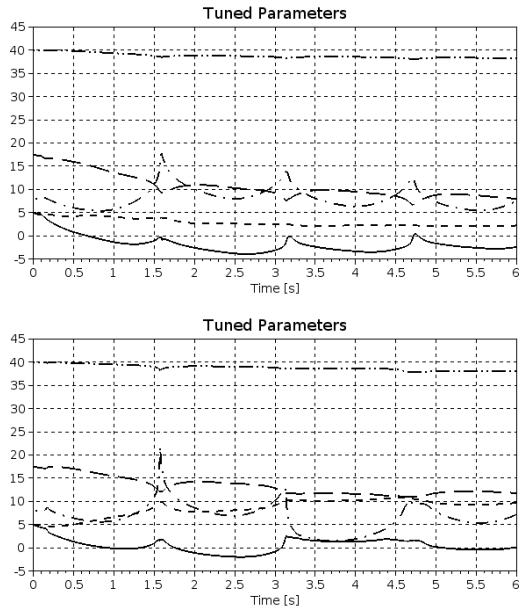

Figure 2.16: Tuning of the adaptive parameters under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[Θ_1: solid, Θ_2: dashed, Θ_3: dense dash, Θ_4: dash-dot, and Θ_5: dash-dot-dot lines]

In this case important details are revealed by the phase trajectories (Fig. 2.17). Without using the RFPT-based adaptation more even and greater tracking errors are present. The RFPT reduces these errors in long sections, while in the problematic sections it generates significant changes in the phase space. Certain details can also be observed in the charts of trajectory tracking (Fig. 2.18). In Fig. 2.19 the operation of the RFPT-based method is illustrated: the realized (simulated) 2^{nd} time-derivative is in the close vicinity of the kinematically computed "desired" value, i.e. the primary design intent, i.e. the realization of a kinemtically prescribed tracking error relaxation is successfully demonstrated.

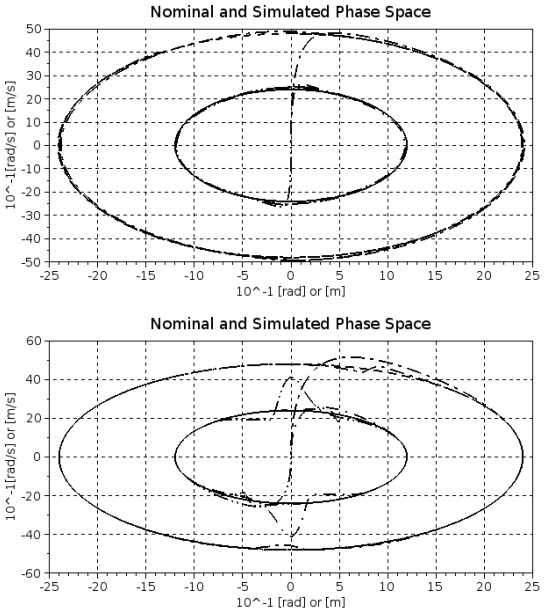

Figure 2.17: The phase trajectories under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines]

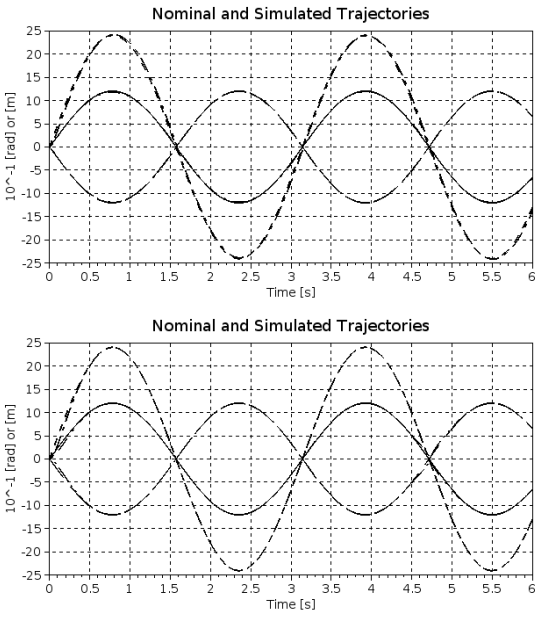

Figure 2.18: The trajectory tracking under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines]

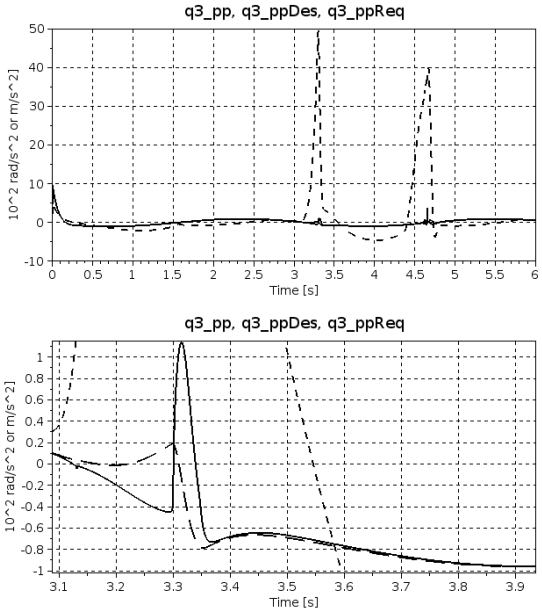

Figure 2.19: The second time-derivatives of generalized coordinate q_3 with modified tuning and RFPT-based adaptation (zoomed excerpt in the lower chart) [$\ddot{q}_3$ (realized): solid, $\ddot{q}_3^{Des}$ ("desired"): dashed, $\ddot{q}_3^{Req}$ (adaptively deformed): dense dash lines]

2.5 Novel Tuning Method for the Modified Adaptive Inverse Dynamic Robot Controller (MAIDRC)

In this section I present that the SGFPT can also well coexist with the MAIDRC control design. I propose a novel, even simpler tuning method that also applies fixed-point transformation based tuning rule for parameter identification [A. 4].

2.5.1 Improvement in the MAIDRC control design

The theoretical considerations are validated by numerical simulations of the adaptive control of a 2 DoF (Degree of Freedom) paradigm. This approach uses the formal properties of the dyanamic model of the system under control as

$$H(q)\ddot{q} + h(q,\dot{q}) \equiv Y(q,\dot{q},\ddot{q})\Theta = Q \ , \tag{2.24}$$

in which $H > 0$ the positive definite symmetric inertia matrix, q is the generalized coordinate of the system, $Y(q,\dot{q},\ddot{q})$ is precisely known on the basis of the *kinematic model* of the controlled system, Q is the exerted generalized control forces, while Θ is the array of the dynamic parameters that are only imprecisely known. Let the approximate values notated as $\hat{H}$, $\hat{h}$, and $\hat{\Theta}$, respectively. The PID-type *kinematic trajectory tracking error relaxation* was prescribed as $\ddot{q}^{Des} = \ddot{q}^{Nom} + 3\Lambda^2(q^{Nom} - q) + \Lambda^3 \int_0^t \left(q^{Nom}(\xi) - q(\xi)\right) \mathrm{d}\xi$. For the *adaptive deformation* the same function was used here, as in [A. 10]:

$$\begin{gathered} F(x) \stackrel{def}{=} \operatorname{atanh}(\tanh(x+D)/2) \ , \\ h_i \stackrel{def}{=} f(r_i) - r_{i+1}^{Des} \ , \\ e_i \stackrel{def}{=} \tfrac{h_i}{\|h_i\|} \ , \\ r_{i+1} = G\left(r_i, f(r_i), r_{i+1}^{Des}\right) \stackrel{def}{=} \\ \left[F\left(A\,\|h_i\| + x_\star\right) - x_\star\right] e_i + r_i \ , \end{gathered} \tag{2.25}$$

where $F(x_\star) = x_\star$, and the Frobenius norm is in use. The applied new type of function is detailed in section 4.2. Let the exerted control force calculated as

$$Q_n = \hat{H}(q_n)\,[r_n] + \hat{h}(q_n,\dot{q}_n) \ , \tag{2.26}$$

that, according to (2.24), must be equal to $H(q_n)\ddot{q}_n + h(q_n,\dot{q}_n)$. By subtracting $\hat{H}\ddot{q} + \hat{h}$ from both sides we get information on the actual modeling errors as

$$\hat{H}(q_n)\,[r_n - \ddot{q}_n] = Y\left(q_n,\dot{q}_n,\ddot{q}_n\right)\left(\Theta - \hat{\Theta}_n\right) \ . \tag{2.27}$$

This structure corresponds to the $a = Yb$ equation with $a_n = \hat{H}(q_n)\left[r_n - \ddot{q}_n\right]$ and $b_n = \left(\Theta - \hat{\Theta}_n\right)$. The novelty in this approach that for *parameter tuning* I replace the relatively complicated *Modified Gram-Schmidt Algorithm* by a more primitive iterative approach [A. 4]: we have information on the parameter estimation error in the orthogonal sub-space of the parameter space spanned by the *rows* of Y, therefore we seek this part of the error in the form $\left(\Theta - \hat{\Theta}_n\right) = Y^T\left(q_n, \dot{q}_n, \ddot{q}_n\right) w_n \equiv Y_n^T w_n$. Within each control cycle a quick internal iteration is run for finding the appropriate w_n parameter as [A. 4]:

$$w_n(i+1) = G\left(w_n(i), Y_n Y_n^T w_n(i), a_n\right) \quad . \tag{2.28}$$

Since $Y_n Y_n^T$ is symmetric positive semidefinite, convergence can be guaranteed for it for positive eigenvalues, and signal stagnation for a zero eigenvalue, as it was used for inverese kinematic applications in [49]. For the purposes of the *dynamic control* and the *parameter tuning* different parameter A can be chosen in (3.5). Furthermore, the internal iteration in the first step can be commenced at an arbitrary value in the first control cycle, and in the forthcoming ones from the value at which the iteration was stopped in the previous cycle. In this manner very low computational need can be expected. The parameter tuning was applied for the known components as $\dot{\hat{\Theta}} = -\alpha\left(\Theta - \hat{\Theta}\right)$ with a parameter $\alpha > 0$ [A. 4].

2.5.2 Simulation Results

The applicability has been validated by numerical simulations made for a 2 *Degree of Freedom (DoF)* paradigm, in the adaptive control of two coupled mass-points with simultaneous parameter identification. The system consist of two masses ($m_1\,[kg]$, $m_2\,[kg]$) connected with two springs, the zero force lengths of which are denoted by $L_1\,[m]$ and $L_2\,[m]$. The spring stifnesses are $k_1\,[N/m]$ and $k_2\,[N/m]$. The viscous damping coefficients are denoted by $b_1\,[N \cdot s/m]$ and $b_2\,[N \cdot s/m]$. The equation of motion is given by:

$$\begin{gathered}\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + h = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \quad , \\ h = \\ \begin{bmatrix} -m_1 g + k_1(q_1 - L_1) - k_2(q_2 - q_1 - L_2) + b_1\dot{q}_1 \\ -m_2 g + k_2(q_2 - q_1 - L_2) + b_2\dot{q}_2 \end{bmatrix}\end{gathered} \tag{2.29}$$

The task is to adjust the necessary forces $Q_1\,[N]$ and $Q_2\,[N]$ in order to allow the desired displacement of the two masses $q_1\,[m]$ and $q_2\,[m]$.

The initial values for the approximate system parameters were

$$\hat{\Theta}_0 = \\ \left[m_{1a}, m_{1a}g, k_{1a}, k_{1a}L_{1a}, k_{2a}, k_{2a}L_{2a}, b_{1a}, m_{2a}, m_{2a}g, b_{2a}\right]^T .$$

The controller's parameters were stated in $D = 0.3$, $A_1 = -1$, $A_2 = -3$ $\alpha = -3e - 1$ and the numerical Euler integration was carried out with $\delta t = 0.001$ with a time constant $\Lambda = 5\,[s^{-1}]$. Figs. 2.20 - 2.21 shows the trajectory tracking while Fig. 2.22 and Fig. 2.23 display the trajectory tracking errors. It can be seen, that compassing the iteration from the value 0 (Fig. 2.22) ensures nearly the same smooth performance as using the cycles' past results as initial values 2.21. Similar results for the phase trajectory can be seen in Figs. 2.24 - 2.25. The errors of the approximation of the known components in Figs. 2.26 - 2.27 show that in the second case the error decreases faster. The tuning of the dynamic parameters are revealed in Figs. 2.28 - 2.31. It can be seen that the learning speed is faster in the second case for most of the parameters, such as parameters $\hat{\theta}_1, \hat{\theta}_4, \hat{\theta}_5, \hat{\theta}_9, \hat{\theta}_{10}$. Figs. 2.32 - 2.33 reveal the performance of the adaptivity. After switching the adaptivity on at $t = 0.5[s]$ it can be observed that $\ddot{q}_1$ approaches $\ddot{q}_1^{Des}$ and $\ddot{q}_2$ approaches $\ddot{q}_2^{Des}$ and the deformed signals separated.

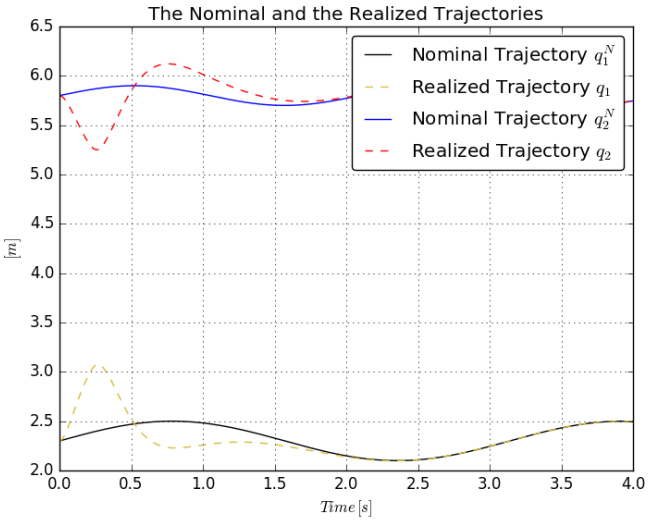

Figure 2.20: The Trajectory Tracking (iteration using initial value 0)

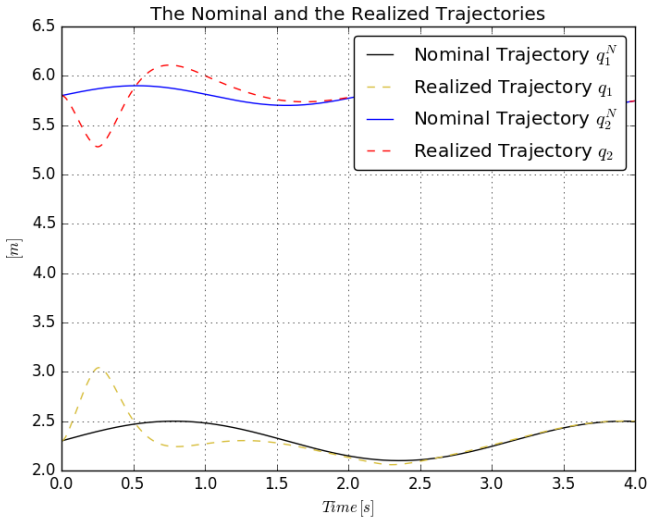

Figure 2.21: The Trajectory Tracking (using result of previous cycle as initial value)

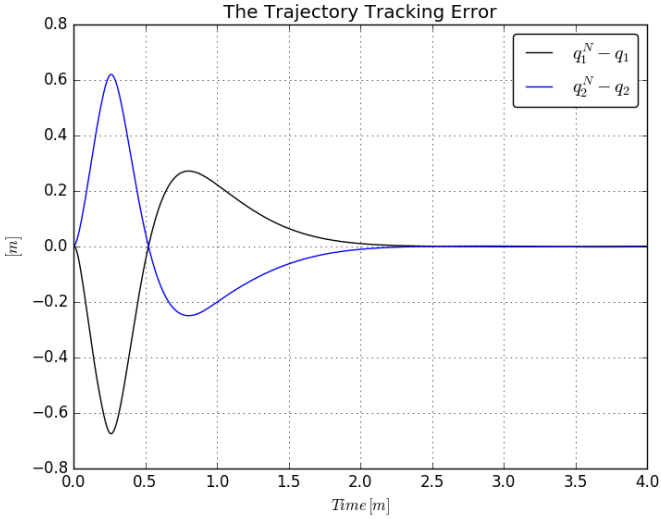

Figure 2.22: The Trajectory Tracking Error (iteration using initial value 0)

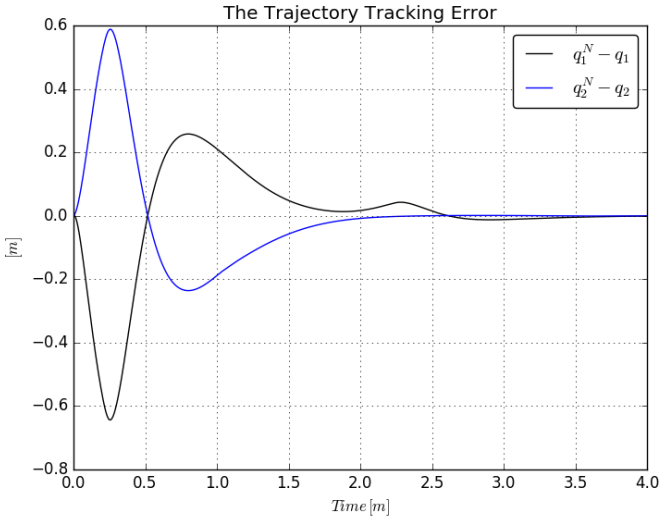

Figure 2.23: The Trajectory Tracking Error (using result of previous cycle as initial value)

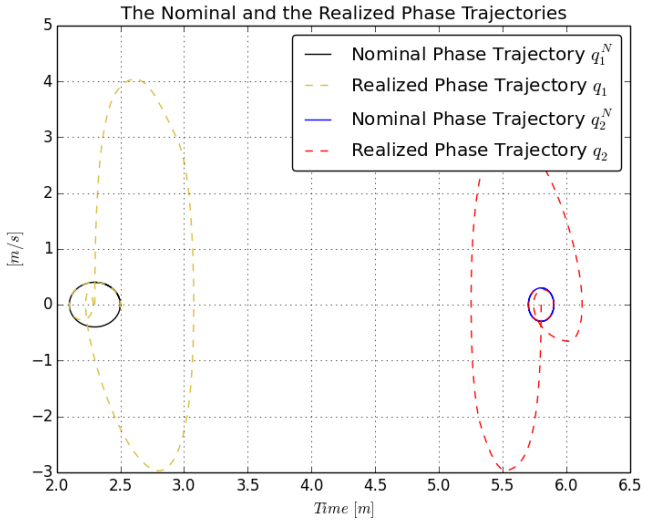

Figure 2.24: The Phase Trajectory Tracking (iteration using initial value 0)

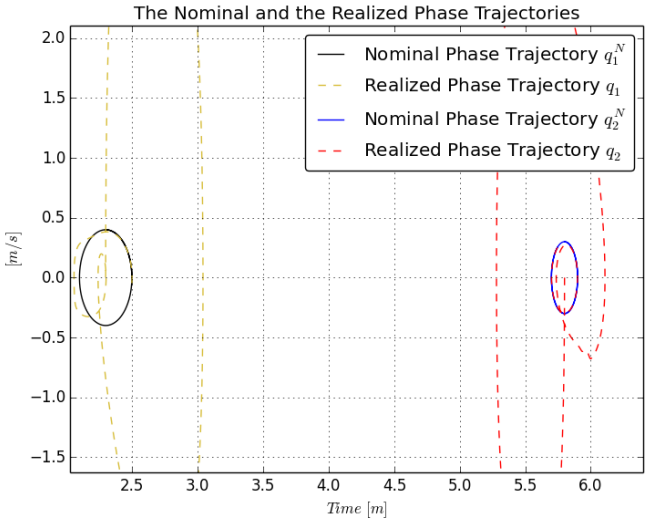

Figure 2.25: The Phase Trajectory Tracking (using result of previous cycle as initial value)

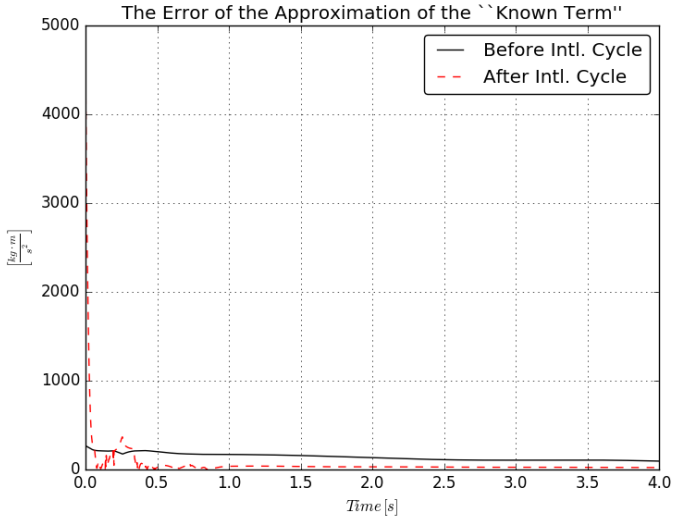

Figure 2.26: The Error of the Known Term (iteration using initial value 0)

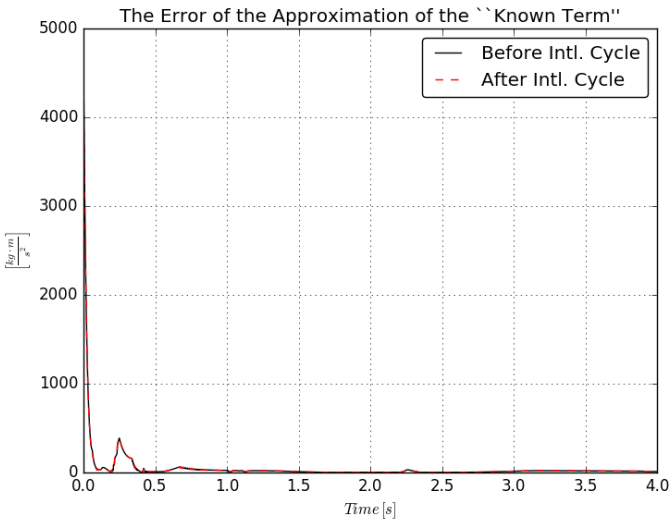

Figure 2.27: The Error of the Known Term (using result of previous cycle as initial value)

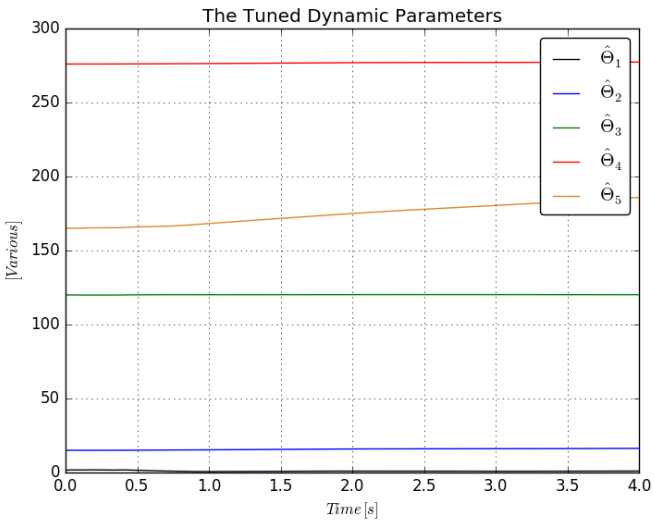

Figure 2.28: The Dynamic parameters (iteration using initial value 0) $\hat{\theta}_1 - black, \hat{\theta}_2 - blue, \hat{\theta}_3 - green, \hat{\theta}_4 - red, \hat{\theta}_5 - orange$

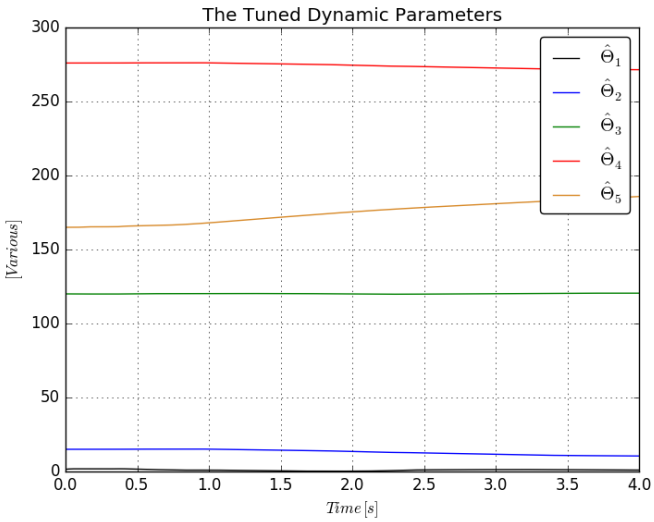

Figure 2.29: The Dynamic parameters (using result of previous cycle as initial value) $\hat{\theta}_1 - black, \hat{\theta}_2 - blue, \hat{\theta}_3 - green, \hat{\theta}_4 - red, \hat{\theta}_5 - orange$

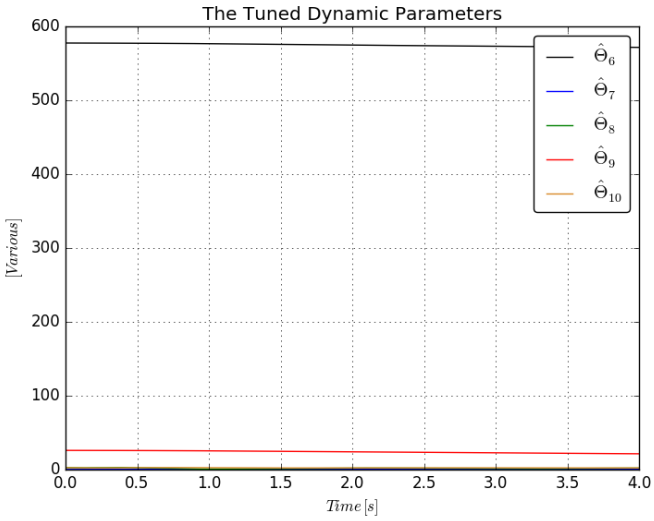

Figure 2.30: The Dynamic Parameters (iteration using initial value 0) $\hat{\theta}_6 - black, \hat{\theta}_7 - blue, \hat{\theta}_8 - green, \hat{\theta}_9 - red, \hat{\theta}_{10} - orange$

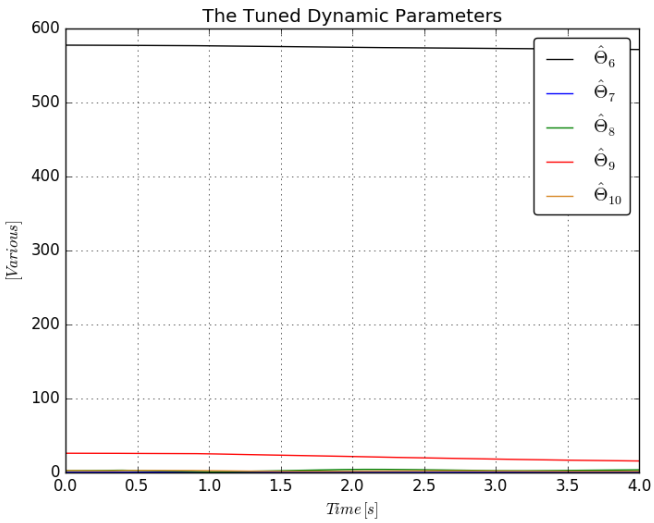

Figure 2.31: The Dynamic parameters (using result of previous cycle as initial value) $\hat{\theta}_6 - black, \hat{\theta}_7 - blue, \hat{\theta}_8 - green, \hat{\theta}_9 - red, \hat{\theta}_{10} - orange$

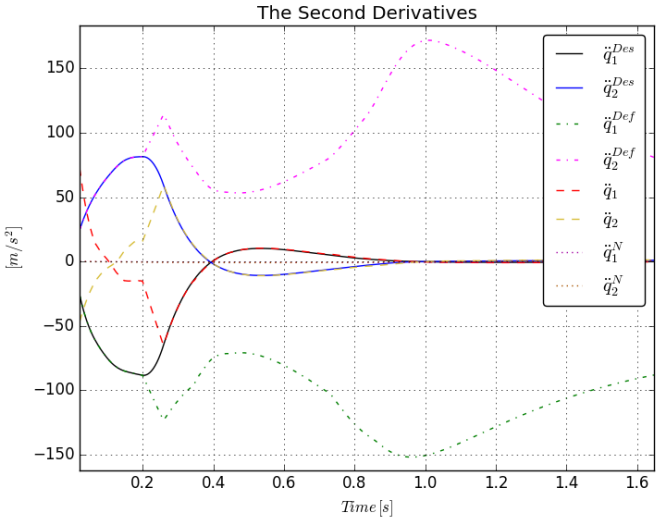

Figure 2.32: The 2^{nd} Time Derivatives (iteration using initial value 0)

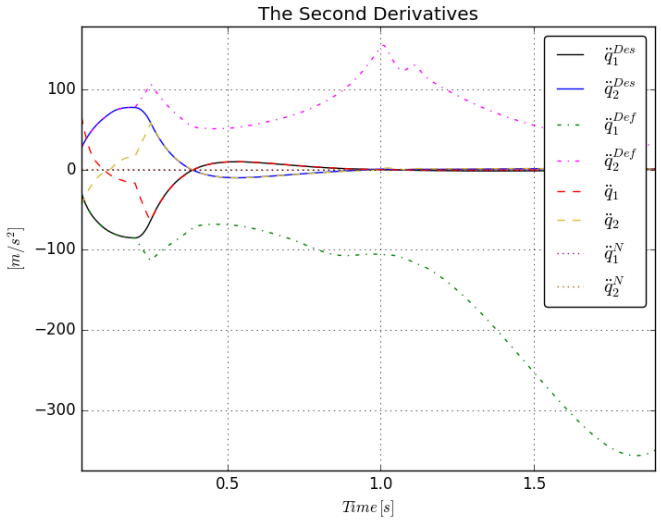

Figure 2.33: The 2^{nd} Time Derivatives (using result of previous cycle as initial value)

2.6 Thesis Statement I.

I have introduced a new method for the combination of the classical model-based adaptive control approaches as the "Modified Adaptive Inverse Dynamics Robot Controller" (MAIDRC) and the "Modified Adaptive Slotine-Li Robot Controller" (MADSLRC) with the Robust Fixed Point Transformation-based design. The proposed new algorithm ensures efficient parameter identification by the use of the modified Gram-Schmidt method. Additionally, I have developed a new, even more simplified strategy that also applies fixed point transformation-based tuning rule in the case of the MAIDRC control design.

Thesis Explanation

The key point of this new concept relies on the fact that the original initial equations of motion can be so rearranged, that instead serving the need of creation a quadratic Lyapunov function, direct information can be obtained on a "fragment" of the actual model parameter estimation error. On the basis of a simple geometric interpretation this information can be directly used for parameter tuning. The modified Gram-Schmidt algorithm is proposed in the parameter tuning process on the matrix of the kinematically known dynamic system parameters that yields more efficient tuning and in general requires less number of arbitrary control parameters. This step greatly simplifies the calculations. In contrast to the traditional solutions that normally guarantee global (asymptotic) stability by using Lyapunov functions, the adaptive controllers designed by the use of "Robust Fixed Point Transformations (RFPT)" are only locally stable, cannot learn the system's analytical model parameters but they are very robust to modeling deficiencies (for instance, abandoned friction effects) and unknown external forces. Utilizing the fact that the mathematical form of the new rearrangement also allows the use of the RFPT-based iterative feedback terms, I have shown that the learning ability of the original controllers can be efficiently combined with the robustness of the RFPT-based feedback in the lack of unknown external disturbances. I have also shown that though the unknown external disturbances corrupt the parameter tuning in this combined solution, the RFPT-based feedback efficiently can compensate the effects of temporal external perturbations of short duration. Furthermore, I have designed a new fixed point transformation-based parameter identification strategy for the Modified Adaptive Inverse Dynamics Robot Controller. The proposed new algorithm is far simpler and has lower computational need. I have illustrated these effects via numerical simulations. Thesis application and proof of operation can be found in 2.3.1, 2.4.4.
Related publications: [A. 1], [A. 3], [A. 2], [A. 4]

Chapter 3

New Generation of Fixed Point Transformation for Adaptive Control

3.1 Fixed Point's Generation for SISO Systems

The previous investigations belong to a special fixed point transformation the so-called RFPT. The question naturally arose: is it possible to construct other type fixed point transformations that are applicable for adaptive control? In this Chapter a positive answer is given to this question.

3.1.1 The Idea of Fixed Point Generation

The idea is borrowed from the subject area of fuzzy operators. In fuzzy controllers the so-called *t-norms* and *t-conorms* can be built up by the use of monotonic generator functions $\varphi(x) : \mathbb{R} \mapsto \mathbb{R}$ as $\psi(x,y) \stackrel{def}{=} \varphi\left(\varphi^{(-1)}(x) + \varphi^{(-1)}(y)\right)$ [50]. In signal aggregation a similar technique can be used in which the monotonicity of the generator function is of crucial significance [51]. Consider a monotonic increasing bounded, smooth function (a "sigmoid") $g(x) : \mathbb{R} \mapsto \mathbb{R}$. For some $K > 0$ and $D > 0$ consider the graphs of the functions $g(x) - K$ and $g(x - D)$ (Fig. 3.1). In the 1st case the graph of $g(x)$ is shifted vertically (down), while in the second one it is shifted horizontally (to the right). It is evident that generally two points can be obtained, x_{*_1} and $x_{*_2} > x_{*_1}$ for which it is valid that $g(x_{*_1}) - K = g(x_{*_1} - D)$ and $g(x_{*_2}) - K = g(x_{*_2} - D)$ [A. 5].

Since $g(x)$ is invertible there exist the function $F(x) \stackrel{def}{=} g^{-1}\left(g(x) - K\right) + D$, and trivially x_{*_1} and x_{*_2} are the *fixed points* of $F(x)$: $F(x_{*_1}) = x_{*_1}$, and $F(x_{*_2}) = x_{*_2}$. The graph of $F(x)$ is illustrated in Fig. 3.2. It is evident that one of these fixed points is *repulsive*, while the other is *attractive*. Figure 3.3 illustrates that the iteration $x_{n+1} \stackrel{def}{=} F(x_n)$ converges to the

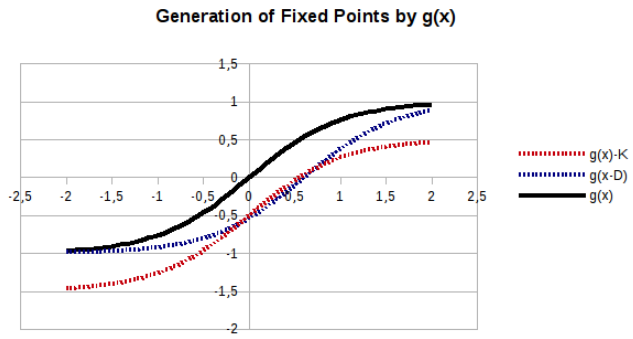

Figure 3.1: The basic idea of fixed point generation by the use of a sigmoid

attractive fixed point if the initial element $x_0 > x_{*_1}$. Let the attractive fixed point be denoted in the sequel as $x_* \equiv x_{*_2}$. The actual value of x_* can easily be determined via iteration.

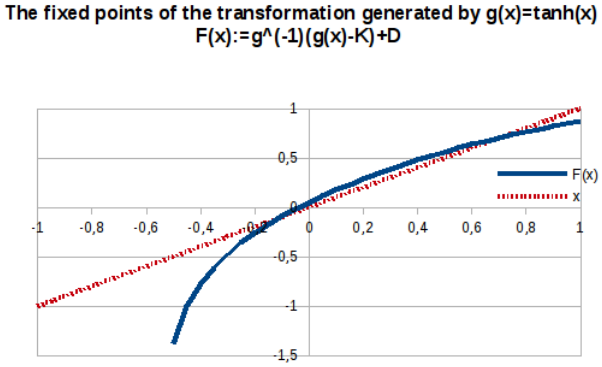

Figure 3.2: The fixed points of $F(x) \stackrel{def}{=} g^{-1}\left(g(x) - K\right) + D$

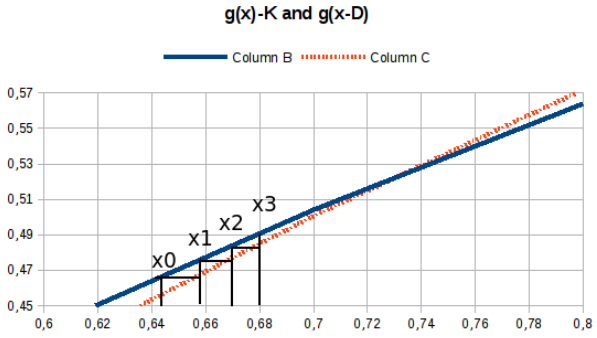

Figure 3.3: Schematic description of the iteration $x_{n+1} \stackrel{def}{=} F(x_n)$ as it converges to the attractive fixed point [$g(x_n) - K = g\left(x_{n+1} - D\right)$ corresponds to $x_{n+1} = g^{-1}\left(g(x) - K\right) + D$]

By the use of $F(x)$ the following fixed point transformation is recommended for adaptive control:

$$r_{n+1} = G(r_n) \stackrel{def}{=} F\left(A\left[f(r_n) - r^{Des}\right] + x_\star\right) + r_n - x_\star \tag{3.1}$$

in which $A \in \mathbb{R}$ is a parameter. Evidently, if $r_\star$ is the solution of the control task, i.e. $f(r_\star) = r^{Des}$ then $G(r_\star) = r_\star$, that is this solution is a fixed point of the function G. For guaranteeing the convergence of the series $\{r_n\}$ function G must be contractive. This contractivity can be achieved by properly setting the value of the parameter A [A. 5]. In the next section typical examples are be considered. The illustrative Figs. 3.1–3.3 belong to the case of $g(x) = \tanh(x)$ with $K = 0.5$ and $D = 0.6$ to which the numerically determined $x_\star \approx x = 0.7114269142$ attractive fixed point belongs. With an *affine approximation* of the *response function* $F(x) = \alpha x + \beta$ with $\alpha = 1$ and $\beta = 1$, for $r^{Des} = -2$ *slow monotonic convergence* can be achieved for $A = -0.5$ (Fig. 3.4) [A. 5].

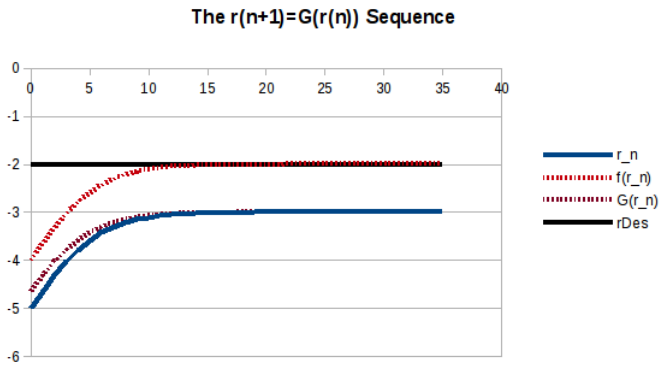

Figure 3.4: Example for slow monotonic convergence to the solution of the control task: $A = -0.5$

Further increase in A at first results in faster convergence (Fig. 3.5), then in the appearance of non-monotonic convergence (Fig. 3.6), finally it leads to bounded chaotic oscillations (Fig. 3.7).

3.1.2 Application Example

Since in [52] the FitzHugh-Nagumo model's adaptive control was investigated by the original RFPT transformation, the same model is chosen to demonstrate the abilities of the novel type fixed point transformation in this section. The *"FiztHugh Neuron Model"* was suggested in 1961 to model the spiking phenomena of neurons [53]. In the next year an equivalent circuit

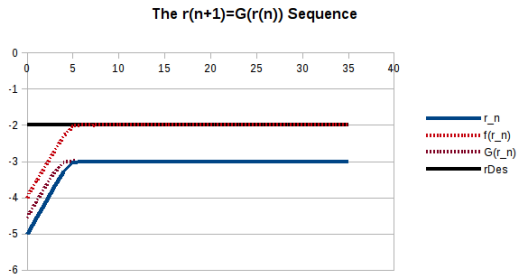

Figure 3.5: Example for fast monotonic convergence to the solution of the control task: $A = -2$

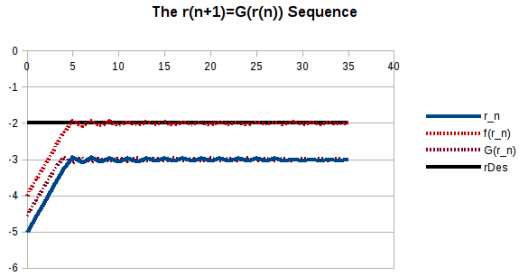

Figure 3.6: Example for precursor oscillations (non-monotonic convergence to the solution of the control task): $A = -3.12$

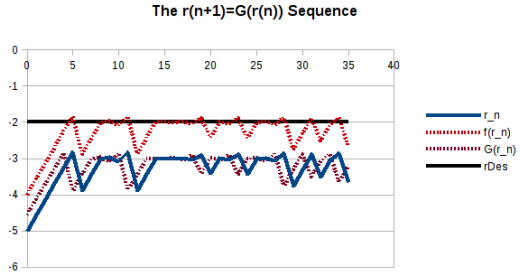

Figure 3.7: Example for chaotic, divergent oscillations: $A = -6$

of this nonlinear oscillator was created by Nagumo et al. in [54]. The equations of motion of this two *Degree of Freedom (DoF)* system are given in (3.2).

$$\frac{\mathrm{d}v}{\mathrm{d}t} = v - \frac{v^3}{3} - w + I_{ext} \tag{3.2a}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{v + a - bw}{\tau} \tag{3.2b}$$

This model has been extended in (3.3) by the introduction of a *new parameter* and an *additional external current* for control purposes.

Parameter	Ideal	Approximate	Actual/Exact
a	$a_i = 0.6$	$a_a = 0.8$	$a_e = 0.7$
τ	$\tau_i = 11$	$\tau_a = 10$	$\tau_e = 12.5$
b	$b_i = 0.4$	$b_a = 0.6$	$b_e = 0.5$
I_{ext}	$I_{ext_i} = 0.3$	$I_{ext_a} = 0.4$	$I_{ext_e} = 0.5$
μ	$\mu_i = 1$	$\mu_a = 3$	$\mu_e = 2$

Table 3.1: The parameters of the trajectory generator "ideal" neuron, that of the "approximate model" and the actually controlled "Actual/Exact" neurons

$$\frac{\mathrm{d}v}{\mathrm{d}t} = v - \frac{v^3}{3} - w + I_{ext} + \mu I_{Ctrl} \tag{3.3a}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{v + a - bw}{\tau} \tag{3.3b}$$

in which I_{Ctrl} denotes the control signal, and μ is a new parameter that weights the control current. In the simulations these models with different parameters are used as is given in Table. 3.1.

In the control task a *"Nominal Trajectory"* as $\{v^N(t), w^N(t)\}$ is generated by the use of the "ideal" neuron's parameter settings with $I_{Ctrl} \equiv 0$. The control signal is generated by the use of the *"Approximate Parameter Settings"*, while the actually controlled system is assumed to differ from both of them with the *"Actual/Exact Parameter Settings"*.

The control task is precise tracking only $v^N(t)$ without concerning the error $w^N(t) - w(t)$ with the following *purely kinematic trajectory tracking error prescription*:

$$\left(\frac{\mathrm{d}}{\mathrm{d}t} + \Lambda\right)^2 \int_0^t \left(v^N(\xi) - v(\xi)\right) \mathrm{d}\xi = 0 \text{ leading to} \tag{3.4a}$$

$$\dot{v}^{Des}(t) = \dot{v}^N(t) + 2\Lambda \left(v^N(t) - v(t)\right) + \Lambda^2 \int_0^t \left(v^N(\xi) - v(\xi)\right) \mathrm{d}\xi \tag{3.4b}$$

with $\Lambda = 0.1\, s^{-1}$. The adaptive parameters are as in the above considerations i.e. $g(x) = \tanh(x)$, $K = 0.5$, $D = 0.6$ with $x_\star = 0.7114269142$. In the adaptive case $A = -0.75$ is set. In the non-adaptive case the necessary derivative calculated by the PID block (3.4) is directly introduced into the approximate model. In the adaptive case it is deformed according to (3.1).

Figures 3.8 and 3.9 reveal that the adaptive controller yields faster error relaxation and far more precise trajectory tracking than the non-adaptive one. The same tendency can be seen in the phase trajectories (Figs. 3.10 and 3.11). The operation of the adaptive controller is

well exemplified by Fig. 3.12 that reveals that due to the adaptive deformation the "Desired" and "Realized" quantities remain in each other's close vicinity, while in the non-adaptive case there is a significant difference between them. Figure 3.13 testifies the great difference between the control currents, too.

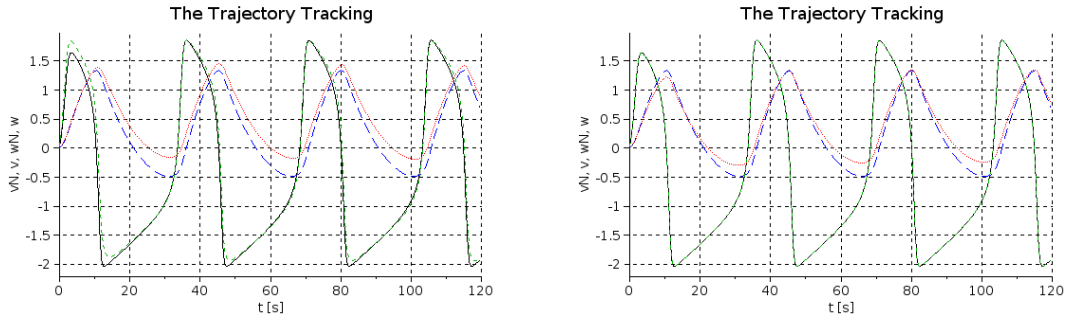

Figure 3.8: Trajectory tracking for v in the non-adaptive (LHS) and the adaptive (RHS) cases: v^N: black solid, w^N: blue dashed, v: green dash dot, w: red dotted lines (LHS)

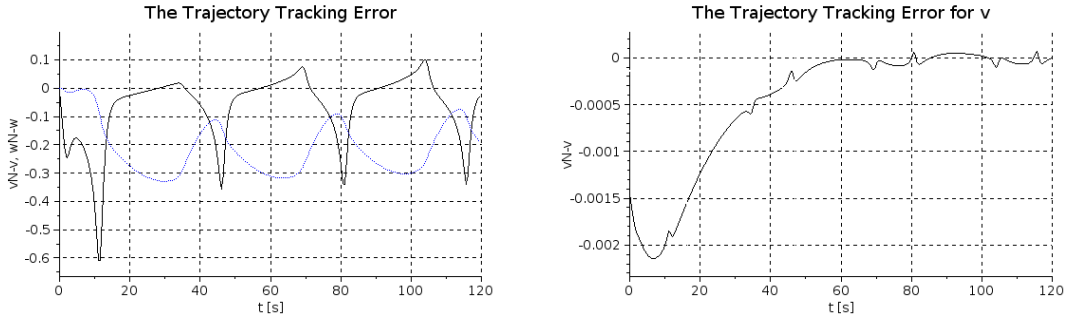

Figure 3.9: Trajectory tracking error for v in the non-adaptive (LHS) and the adaptive (RHS) cases: $v^N - v$: black solid, (in the adaptive case the error of w has been removed from the chart)

Furthermore, the robustness is investigated [A. 5]. The external disturbances and noises are modeled by adding additional sinusoidal and noisy terms to the calculated control signal as follows:

```
if n_evi then
  z=2*(rand(1,1)-0.5)*Ampl_Rand
end
y1=Ampl_Dist*sin(Omega_Dist*t)+z
Ampl_Dist=0.15
Omega_Dist=4
Ampl_Rand=Ampl_Dist
```

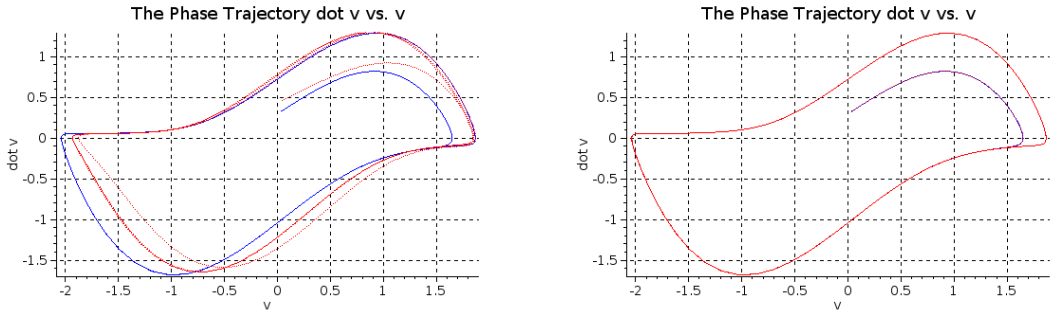

Figure 3.10: The phase trajectories for variable v the non-adaptive (LHS) and the adaptive (RHS) cases: $\dot{v}$ vs. v: nominal: solid blue, realized: red dotted lines

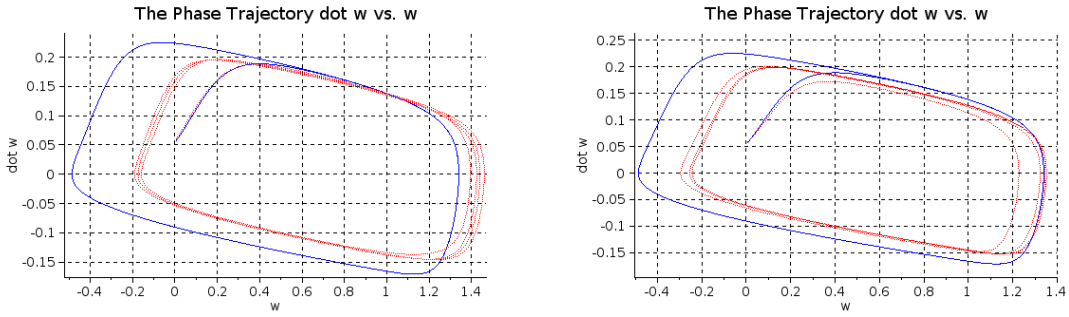

Figure 3.11: The phase trajectories for variable w the non-adaptive (LHS) and the adaptive (RHS) cases when only the trajectory of v is under control: $\dot{w}$ vs. w: nominal: solid blue, realized: red dotted lines

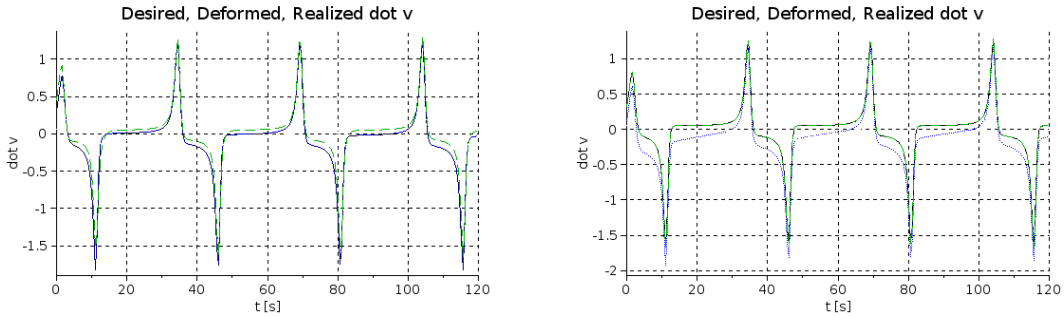

Figure 3.12: The "Desired" (black solid), the "Deformed" (blue dotted) (without deformation it exactly is identical to the "Desired" line), and the "Realized" (green dashed) time-derivatives for variable v in the non-adaptive (LHS), and the adaptive (RHS) cases

In each control cycle the *event clock* generates a signal due to which the noisy term is added to the smooth disturbance term. The results are displayed in Figs. 3.14 and 3.15 revealing that the adaptivity significantly improves the tracking precision even in this case, too. The explanation of this fact is almost trivial: without adaptivity only the PI-type feedback defined

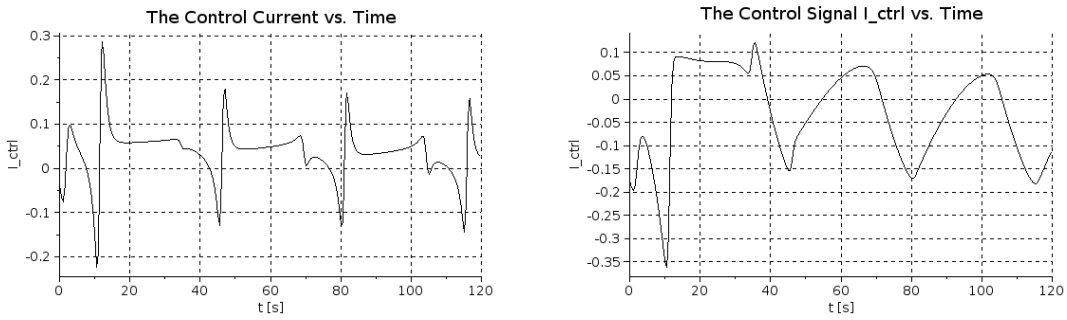

Figure 3.13: The control signal in the non-adaptive (LHS) and the adaptive (RHS) cases

in (3.4) works so that it cannot take into consideration the fast fluctuations in the realized signal $v(t)$. In the adaptive case a fast feedback also works that compensates the fast variation of this signal.

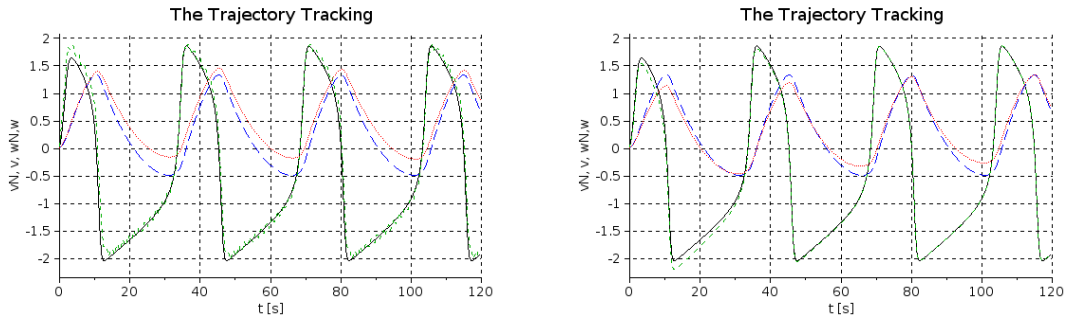

Figure 3.14: Trajectory tracking for v in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: v^N: black solid, w^N: blue dashed, v: green dash dot, w: red dotted lines (LHS)

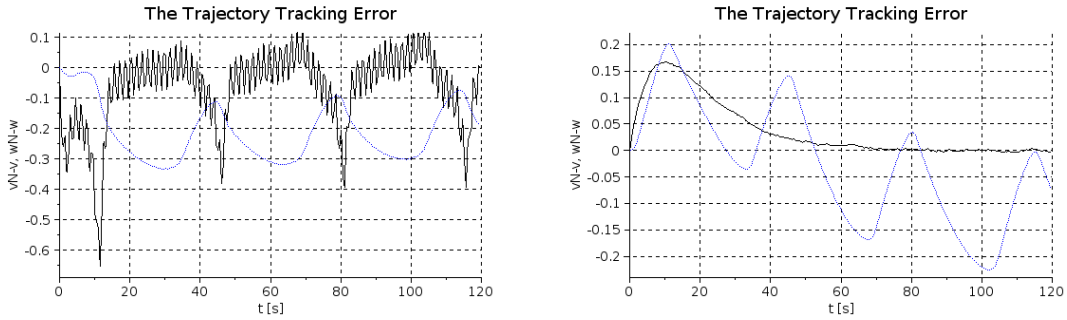

Figure 3.15: Trajectory tracking error for v in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $v^N - v$: black solid, $w^N - w$: blue dotted lines

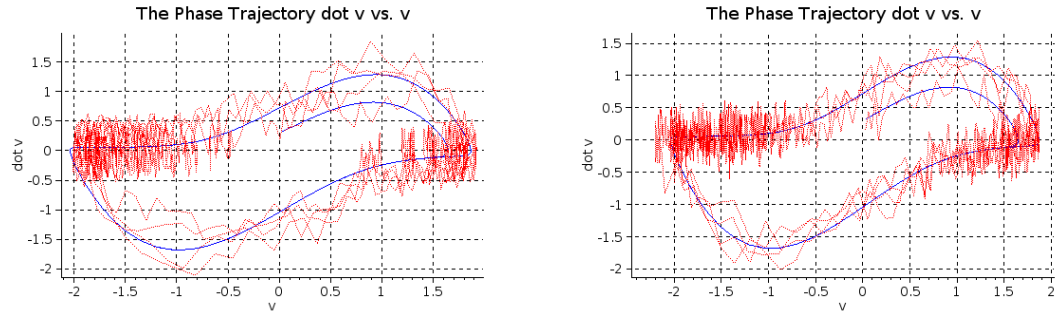

Figure 3.16: The phase trajectories for variable v the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $\dot{v}$ vs. v: nominal: solid blue, realized: red dotted lines

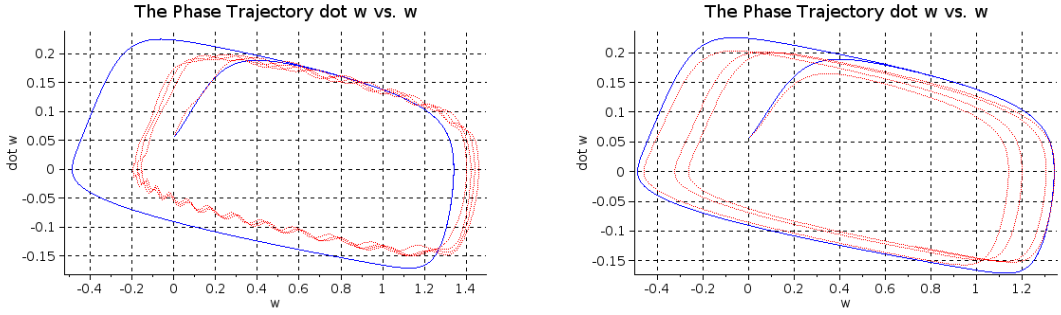

Figure 3.17: The phase trajectories for variable w the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $\dot{w}$ vs. w: nominal: solid blue, realized: red dotted lines

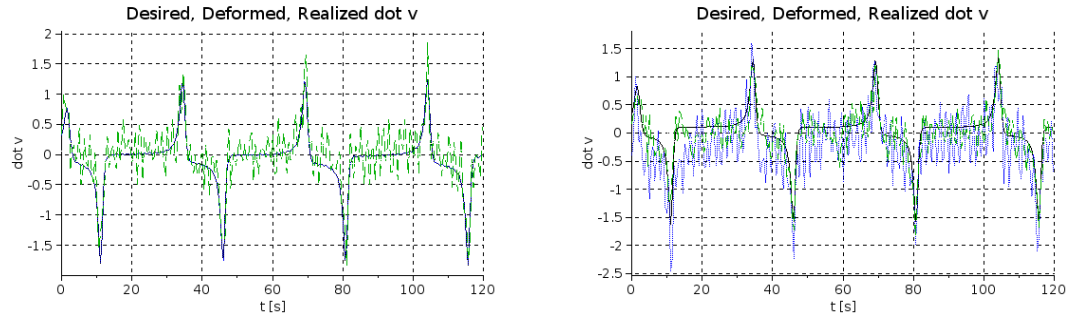

Figure 3.18: The "Desired" (black solid), the "Deformed" (blue dotted) (without deformation it exactly is identical to the "Desired" line), and the "Realized" (green dashed) time-derivatives for variable v in the non-adaptive (LHS), and the adaptive (RHS) cases, under external disturbances.

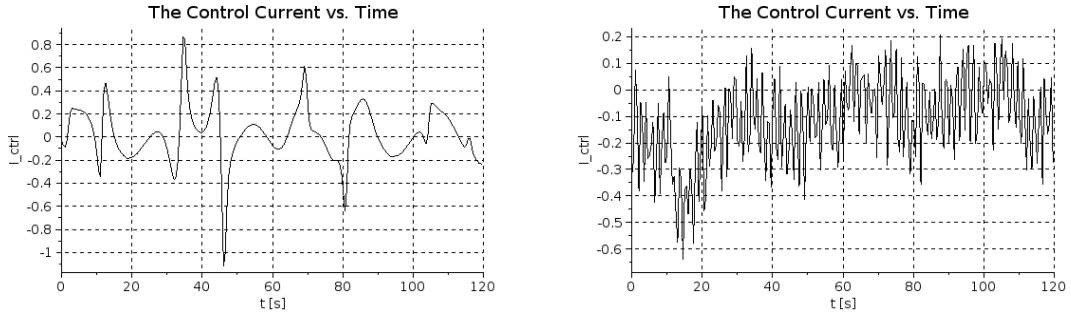

Figure 3.19: The control signal in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances.

3.2 Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO Systems

In the present section I provide further development of this alternative method. The extension of the SGFPT to MIMO systems and its proof is given. The applicability of the novel method is demonstrated by the adaptive control of a 2 "Degree of Freedom (DoF)" system, a cart indirectly driven in the horizontal direction by a rotated pendulum. Results of numerical simulations illustrate and substantiate the usability of the suggested approach.

3.2.1 The Extension to MIMO Systems

A possible extension to MIMO systems when $f, r \in \mathbb{R}^n$, $n \in \mathbb{N}$ is a kind of projection of the nonlinear transformation in the direction of the *response error* in the i^{th} step of the iteration $h(i) \stackrel{def}{=} f(r(i)) - r^{Des}$, $e(i) \stackrel{def}{=} \frac{h(i)}{\|h(i)\|}$ (here the norm $\|h\|$ is understood in the Frobenius sense) [A. 6]:

$$\begin{aligned} r(i+1) = \tilde{G}(r(i)) \stackrel{def}{=} \\ \stackrel{def}{=} \left[F\left(A\,\|h(i)\| + x_*\right) - x_*\right] e(i) + r(i) \ . \end{aligned} \tag{3.5}$$

Remember, that $F(x_*) = x_*$. Evidently if $f(r_\star) - r^{Des} \equiv h(i) = 0$ then $r(i+1) = r(i) = r_\star$, i.e. the solution of the control task is the fixed point of $\tilde{G}(r)$. As is well known, a "smooth" i.e. *infinitely differentiable function* $\Psi(x)$ *as well as its derivatives can be well approximated around a given point* x_0 by their Taylor series expansion as $f(x_0 + \delta x) \approx \sum_{s=0}^{\infty} \frac{f^{(s)}(x_0)}{s!} \delta x^s$ [55]. (In the case of an *analytical function* the series exactly describes the function and its derivatives within the region of convergence.) If δx is small then in this approximation the lowest order terms also give satisfactory approximation. On this reason the first order approximation of function $F(x)$ can be considered in (3.5) around point x_* as [A. 6] :

$$\begin{aligned} r(i+1) \approx \left[F(x_*) + F'(x_*)A\|h(i)\| - x_*\right] \frac{h(i)}{\|h(i)\|} + \\ + r(i) = F'(x_*)Ah(i) + r(i) \end{aligned} \tag{3.6}$$

since $F(x_*) = x_*$. Similar considerations can be applied for function $f(r)$ *in the vicinity of* $r_\star$:

$$\begin{aligned} h \equiv f(r) - r^{Des} = f(r_\star - (r - r_\star)) - r^{Des} \approx \\ \approx f(r_\star) + \left.\frac{\partial f}{\partial r}\right|_{r_\star} (r - r_\star) - r^{Des} = \left.\frac{\partial f}{\partial r}\right|_{r_\star} (r - r_\star) \end{aligned} \tag{3.7}$$

since $f(r_\star) = r^{Des}$. Substituting (3.7) into (3.6) and subtracting $r_\star$ from its both sides the approximation

$$r(i+1) - r_\star \approx \left[I + F'(x_*) A \left. \frac{\partial f}{\partial r} \right|_{r_\star} \right] (r(i) - r_\star) \tag{3.8}$$

is obtained. According to (3.8) it can be stated that $r(i+L) - r_\star \approx (I + \mu M)^L (r(i) - r_\star)$, where for a more concise notation real scalar factor $\mu = F'(x_*)A$, and the matrix $M_{st} = \left. \frac{\partial f_s}{\partial r_t} \right|_{r_\star}$ is introdced. For guaranteeing the convergence of the iteration this sequence must be a Cauchy sequence. For this the function $f(r)$ must have properties that can be considered as follows. In the iteration the consecutive application of (3.8) results in the occurrence of the various powers of the matrix [A. 6] ;

$$(I + \mu M)^m = \sum_{s=0}^{m} \frac{m!}{s!(m-s)!} \mu^s M^s I^{m-s} \ , \tag{3.9}$$

where it is utilized that the identity matrix I commutes with an arbitrary matrix M. To analyze the structure of the powers of M we can refer to the existence of its *Jordan's canonical form* that can be achieved by a *similarity transformation* (see, [56], [57]) transforming M into a *block diagonal structure* as $M = X^{-1} \hat{M} X$ where $\hat{M}$ has block diagonal structure in which the diagonal line just over the main diagonal contains ones, and all the other non-diagonal matrix elements are zeros. Since in the matrix power the block diagonals are not "mixed", and $X^{-1} I X = I$, we have to consider the powers of the blocks of type [A. 6]:

$$\begin{pmatrix} 1+\lambda\mu & \mu & 0 & 0 \\ 0 & 1+\lambda\mu & \mu & 0 \\ 0 & 0 & 1+\lambda\mu & \mu \\ 0 & 0 & 0 & 1+\lambda\mu \end{pmatrix} = (1+\lambda\mu) I + \mu H \tag{3.10}$$

where $\lambda \in \mathbb{C}$ is one of the (normally complex) eigenvalues of M. In an extreme case the size of H may be $n \times n$ (in the case of a single Jordan block) or smaller ($S \times S$, $n > S \in \mathbb{N}$), in the case of the occurrence of more than one Jordan blocks. Matrix H is *nilpotent*, more precisely $H^S = 0$, therefore in (3.9) we have only a limited number of terms even in the case of very big $m \in \mathbb{N}$ powers [A. 6]:

$$[(1+\lambda\mu) + \mu H]^m = \sum_{s=0}^{S-1} \frac{m!}{s!(m-s)!} (1+\lambda\mu)^{m-s} \mu^s H^s \ . \tag{3.11}$$

Equation (3.11) contains the finite number of terms as $(1+\mu\lambda)^m I$, $m(1+\mu\lambda)^{m-1}\mu H$, $\frac{m(m-1)}{2!}(1+\mu\lambda)^{m-2}\mu^2 H^2, \ldots$, and finally

$$\frac{m(m-1)\ldots(m+2-S)}{(S-1)!} (1+\mu\lambda)^{m+1-S} (\mu H)^{S-1} \tag{3.12}$$

in the numerator of which we have $S-1$ terms. The highest power of m that occurs in the last term belongs to $m^{S-1}(1+\mu\lambda)^{m+1-S}$. If $m^{S-1}|1+\mu\lambda|^{m+1-S} \to 0$ as $m \to \infty$ we obtain a Cauchy sequence with the convergence to the solution of the control task $r(i) \to r_\star$. In order to guarantee that, firstly consider the monotonic increasing function $\ln(x)$: $\ln\left(m^{S-1}|1+\mu\lambda|^{m+1-S}\right) = (S-1)\ln(m) + \ln(|1+\mu\lambda|)(m+1-S)$. Since $\frac{\mathrm{d}\ln(m)}{\mathrm{d}m} \to 0$ as $m \to \infty$, the first term has flattening derivative while the second one has a fixed derivative $\ln(|1+\mu\lambda|)$. Therefore if $|1+\mu\lambda| < 1$ then this fixed derivative become negative and $\ln\left(m^{S-1}|1+\mu\lambda|^{m+1-S}\right) \to -\infty$, from which $m^{S-1}|1+\mu\lambda|^{m+1-S} \to 0$ can be concluded. The condition $|1+\mu\lambda| < 1$ is equivalent with $|1+\mu\lambda|^2 < 1$ that means that

$$(1+\mu\Re\lambda)^2 + (\mu\Im\lambda)^2 < 1 \ , \text{ that is} \tag{3.13a}$$

$$1 + 2\mu\Re\lambda + \mu^2|\lambda|^2 < 1 \ , \text{ that is} \tag{3.13b}$$

$$2\mu\Re\lambda + \mu^2|\lambda|^2 < 0 \tag{3.13c}$$

must be valid [A. 6]. For $\Re\lambda > 0$ evidently $\mu < 0$ is needed for that as

$$\frac{-2\Re\lambda}{|\lambda|^2} < \mu < 0 \ , \tag{3.14}$$

and for $\Re\lambda < 0$ evidently $\mu > 0$ is needed for that as

$$0 < \mu < \frac{-2\Re\lambda}{|\lambda|^2} \ . \tag{3.15}$$

These conditions must be valid for each eigenvalue of M. Since $\mu \in \mathbb{R}$ is a single number, it is evident that if $\forall i \ \Re\lambda_i > 0$ or $\forall i \ \Re\lambda_i < 0$ the conditions of the convergence can be met. However, if for certain eigenvalues $\Re\lambda_i > 0$ and for others $\Re\lambda_j < 0$ the contractivity cannot be guaranteed.

Regarding the practical significance of the condition of convergence consider fully driven robots with the equation of motion (for example, [58]) $Q = H(q)\ddot{q} + h(q,\dot{q})$. If the *approximate model* has the terms $\tilde{H} = H + \Delta H$, $\tilde{h} = h + \Delta h$, for the exerted generalized force Q it will be valid that $Q = \tilde{H}\ddot{q}^{Des} + \tilde{h} = H\ddot{q} + h$ from which the response function

$$\ddot{q} = \left(I + H^{-1}\Delta H\right)\ddot{q}^{Des} + H^{-1}\Delta h \tag{3.16}$$

is obtained with $M = I + H^{-1}\Delta H$. For not too drastic modeling error our restriction for the spectrum of M is realistic.

To exemplify the operation of this MIMO extension the parameters belonging to Fig. 3.1 is considered to which $x_* \approx 0.7114269142$ belongs. An affine system model for the response function given in (3.17) is considered with various parameters A [A. 6]. In this model x_1 and

x_2 are evidently coupled. The "desired response" is $[1, -3]^T$. The results can be seen in Fig. (3.20), that exemplify the operation of the MISO extension.

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} 3.832 & 1.819 \\ 1.819 & 2.168 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{3.17}$$

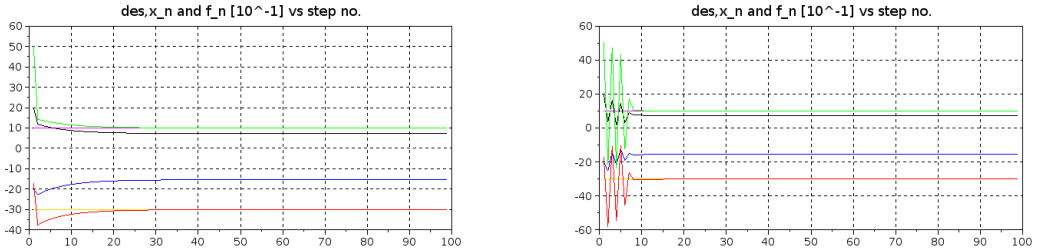

Figure 3.20: The convergence to the desired values for $A = 0.20$ (at the LHS) and $A = 0.27425$ (at the RHS): f_1^{Des}: magenta, f_2^{Des}: ocher, x_1: black, x_2: blue, f_1: green, f_2: red lines

In the next section simulation examples are shown for the partly passively driven Classical Mechanical system that has been considered in [43] and [A. 8].

3.2.2 Application Example

The "*TORA*" (*Translational Oscillations with an Eccentric Rotational Proof Mass Actuator*) corresponds to a simplified model of a dual-spin spacecraft with mass imbalance therefore it serves as a "benchmark problem" for controller design in various publications [59]. For instance in [60] it has been controlled by a cascade and a passivity based controller, while in [61] the "*Tensor Product Form*" of the system model has been applied to develop a model-based controller. In [62] nine papers can be found on the control of the TORA system in a special issue.

The here considered model is an extension of this system to a 3 DoF model in its fully driven form. The system consists of a cart moving in the horizontal direction (generalized coordinate $q_3\,[m]$) with the generalized force $Q_3\,[N]$. To the cart body a pendulum is attached with a rotary joint (coordinate $q_1\,[rad]$) with the driving torque $Q_1\,[N \cdot m]$). At the end of the pendulum a dial can be rotated (coordinate $q_2\,[rad]$) with the driving torque $Q_2\,[N \cdot m]$). In the *underactuated version* $Q_3 \equiv 0$, for q_3 and q_2 we can prescribe *"nominal trajectories"* by allowing the appropriate motion for q_1 and exerting the driving torque values Q_1 and Q_2 according to the equation of motion in Eq. (3.18).

$$\begin{bmatrix} -\frac{\left(mL^2+\Theta\right)(m+M)}{mL\cos q_1}+mL\cos q_1 & \Theta \\ -\frac{(m+M)\Theta}{mL\cos q_1} & \Theta \end{bmatrix} \begin{bmatrix} \ddot{q}_3 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} \frac{\left(mL^2+\Theta\right)mL\sin q_1 \dot{q}_1^{\,2}}{mL\cos q_1} - mLg\sin q_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{3.18}$$

The *exact model parameters* are $m = 20\,[kg]$ (the mass of the dial), $M = 30\,[kg]$ (the mass of the body of the cart, $L = 2\,[m]$ (the length of the beam of neglected mass), and $\Theta = 20\,[kg \cdot m^2]$ (the momentum of inertia of the dial with respect to its own mass center point). The *approximate model parameters* are as follows: $\tilde{m} = 10\,[kg]$, $\tilde{M} = 20\,[kg]$, $\tilde{L} = 2\,[m]$, and $\tilde{\Theta} = 5\,[kg \cdot m^2]$. For the *kinematic trajectory tracking* the $\ddot{q}_i^{Des} = \ddot{q}_i^{Nom} + 3\Lambda^2(q_i^{Nom} - q_i) + 3\Lambda(\dot{q}_i^{Nom} - \dot{q}_i) + \Lambda^3 \int_0^t \left(q_i^{Nom}(\tau) - q_i(\tau)\right) d\tau$ with $\Lambda = 6\,[s^{-1}]$ is prescribed for $i = 2, 3$. The nominal trajectory is a 3^{rd} order periodic spline function of the time resulting "linear" segments in $\ddot{q}_i^{Nom}$.

The cycle time of the digital controller is assumed to be $\delta t = 10^{-3}\,[s]$. In the *adaptive controller* the fixed setting $A_c = -0.1$ is applied. In Fig. 3.21 the details of the trajectory tracking and in Fig. 3.22 the trajectory tracking errors (using the same scaling in the charts) are given. The phase trajectories can be monitored in Fig. 3.23. These figures reveal that the suggested adaptive approach significantly improves the trajectory and phase trajectory tracking properties of the controller.

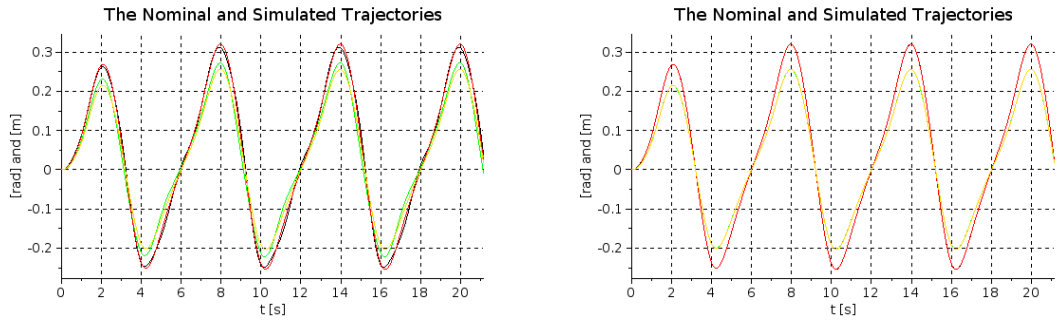

Figure 3.21: Trajectory tracking in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green, q_2^{Nom} $[rad]$: red, q_3^{Nom} $[m]$: ocher lines]

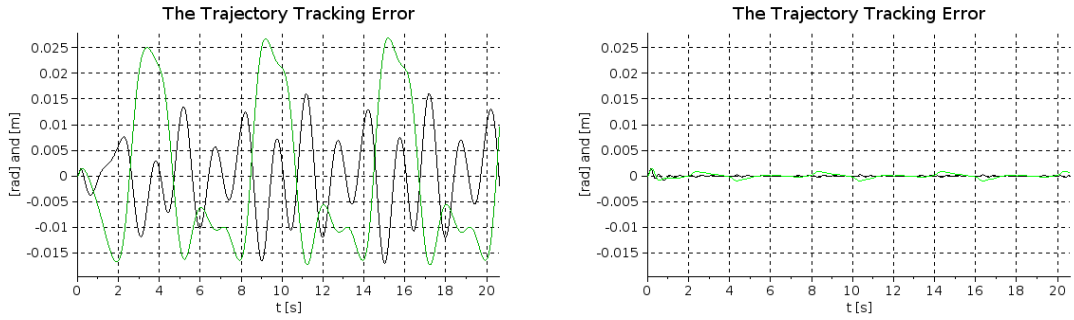

Figure 3.22: Trajectory tracking error in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green lines]

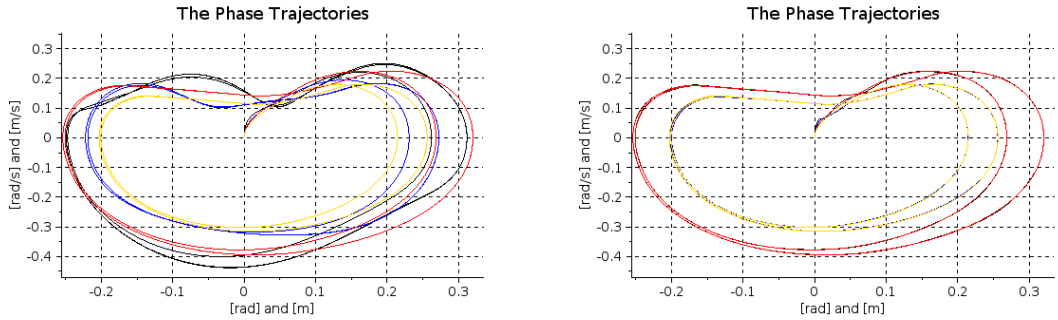

Figure 3.23: The phase trajectories for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [for q_2: black, q_3: blue, q_2^{Nom}: red, q_3^{Nom}: ocher lines]

The operation of the adaptivity can well be seen in Figs. 3.24 and 3.25: due to the adaptive deformation of the input the $\ddot{q}_2^{Des}$: black and $\ddot{q}_2$: brown lines are in each other's close vicinity. The same holds for the $\ddot{q}_3^{Des}$: green, and $\ddot{q}_3$: blue lines. The deformed $\ddot{q}_2^{Def}$: red, and $\ddot{q}_3^{Def}$: ocher lines are considerably different to their counterparts. Subtle differences can also be observed in Fig. 3.26 in which the trajectory of the driving arm q_1 is described.

As it theoretically was expected some increase in the absolute value of A (in this case A varied from -0.1 to -3.125) still improves the precision (Fig. 3.27).

The simulation results reveal that somewhere between $A = -3.23$ and $A = -3.235$ in the control signal very quickly strong chattering appears. This is definitely not desirable for a real control. It anticipates, that in contrast to the original RFPT transformation that was applied in [43], where slow appearance of precursor oscillations were observed, this novel fixed point transformation is apt to turn into an oscillating regime very quickly.

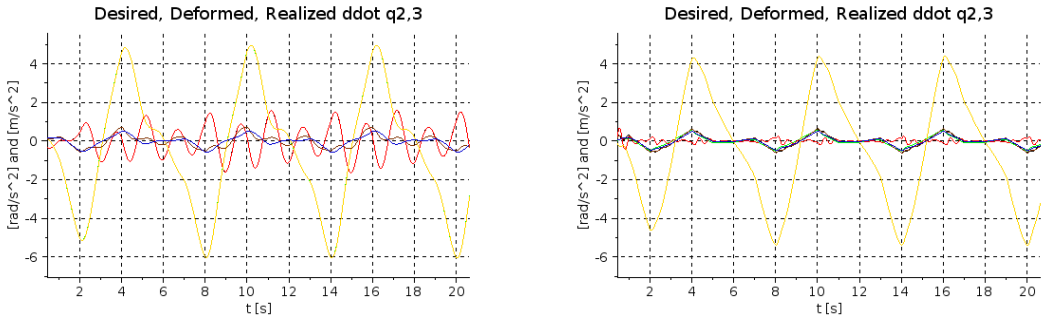

Figure 3.24: The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines]

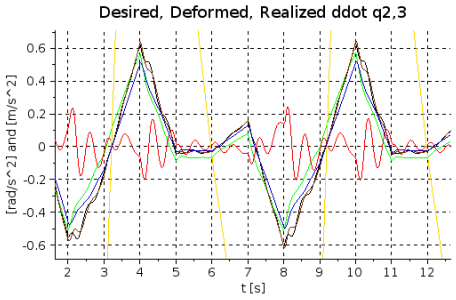

Figure 3.25: The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines]

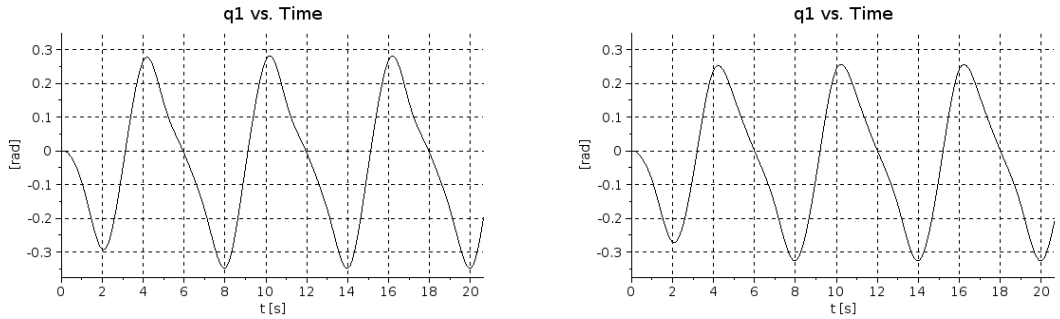

Figure 3.26: The trajectory of the "driving arm" q_1 for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases

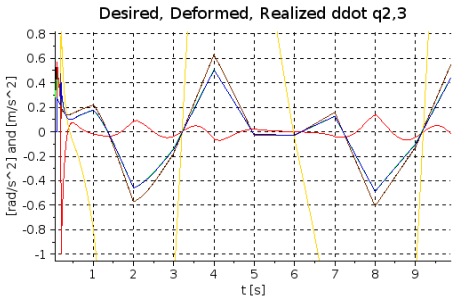

Figure 3.27: The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) for $A = -3.125$ [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines]

3.3 New Advances Regarding the Parameter Tuning

It was shown in the latter section that a simple tuning of parameter A, as it was done in the case of the original RFPT-based method, is not expedient, because the system very quickly takes strongly oscillatory behavior. Thus, in the following I study this critical "transient" region via simulations and theoretical considerations.

3.3.1 Replacement of Parameter Tuning with Simple Calculation

It became clear that A can vary in a wide range within which it considerably improves the quality of the control with respect to the non-adaptive one. A slow increase in $|A|$ improves the precisison till the appearance of the strong fluctuation in the control signal. This observation generated the idea as follows [A. 7]:

- It is not necessary to well approximate the upper limit of $|A|$ to achieve good control quality.
- It is just enough to set $|A|$ so that the argument of the function $F(A\|f - r^{Des}\|)$ is located somewhere between the two fixed points that can be easily determined even graphically by considering the graph in Fig. 3.1.
- On this basis a *"width parameter"* $W \stackrel{def}{=} 0.2 \times (0.7 - (-0.1))$ can be introduced and it has to be achieved that $|A| \cdot \|f - r^{Des}\| \approx W$
- It is important to note that f is known via the observation of the system response and r is known from the control signal.

To realize the above idea, for the control cycle time δt and a forgetting factor $\beta \in (0, 1)$ the actualized value of the response error at cycle n is [A. 7]:

$$\bar{h}(n) \stackrel{def}{=} (1 - \beta) \sum_{s=0}^{\infty} \beta^s \|h(n - s)\| \ . \tag{3.19}$$

For a constant signal $\|h\|$ (3.19) yields $\bar{h}(n) = \|h\|$ therefore (3.19) corresponds to a weighted sum of the present and past values of $\|h(i)\|$. For small possible β the relative weight of the past values is small (quick forgetting) while for greater β this forgetting is not so fast. In the possession of this value the actual value for $|A|$ can be set as [A. 7],

$$A_{Act}(n) \stackrel{def}{=} \frac{W}{\varepsilon + \bar{h}(n)} \ , \tag{3.20}$$

where $\varepsilon > 0$ is a small positive number to avoid division by zero. This estimation is far simpler than the formerly used tuning for the appropriate counterpart of the present parameter A. Its digital realization is very simple: in each control cycle a buffer's content can be refreshed as $b \leftarrow \beta b + \|h\|$, and the output is $(1 - \beta)b$.

3.3.2 Application Example

For the simulation results the same TORA system is used, that is described with the equation of motion in Eq. (3.18). In the *adaptive controller* the controller cycle time is $\delta t = 10^{-3}\,[s]$, the forgetting factor is $\beta = 0.99$ and the value of A_c is calculated according to (3.20). The results on the trajectory tracking and the trajectory tracking errors can be observed in Fig. 3.28 and Fig. 3.29, while Fig. 3.30 shows the phase trajectories. These figures clearly display the performance improvement by the proposed alternative approach comapared to the non-adaptive case. The simulation results demonstrate also, that we have obtained accurate trajectory and phase trajectory tracking with the replacement of parameter tuning with the proposed simple calculation technique.

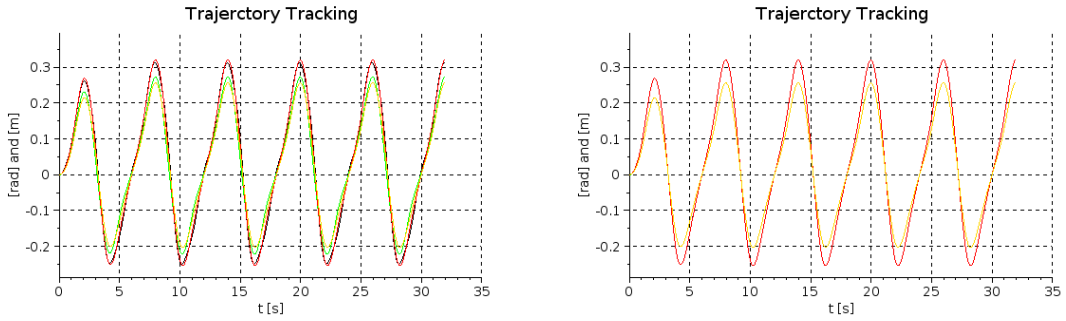

Figure 3.28: Trajectory tracking in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [$q_2\ [rad]$: black, $q_3\ [m]$: green, $q_2^{Nom}\ [rad]$: red, $q_3^{Nom}\ [m]$: ocher lines]

Figs. 3.31 and 3.32 detail the operation of the adaptivity due to the adaptive deformation of the input. The desired $\ddot{q}_2^{Des}$ which is represented with black line and $\ddot{q}_2$ with brown line are covering each other. Similar results are obtained for the green line of $\ddot{q}_3^{Des}$ and blue line of $\ddot{q}_3$. The deformed $\ddot{q}_2^{Def}$ red, and $\ddot{q}_3^{Def}$ ocher lines significantly differ from each other. In Fig. 3.33 comparing the trajectory of the driving arm q_1 in adaptive and non-adaptive case only insignificant differences can be recognized. The calculated parameter A is shown in Fig. 3.34.

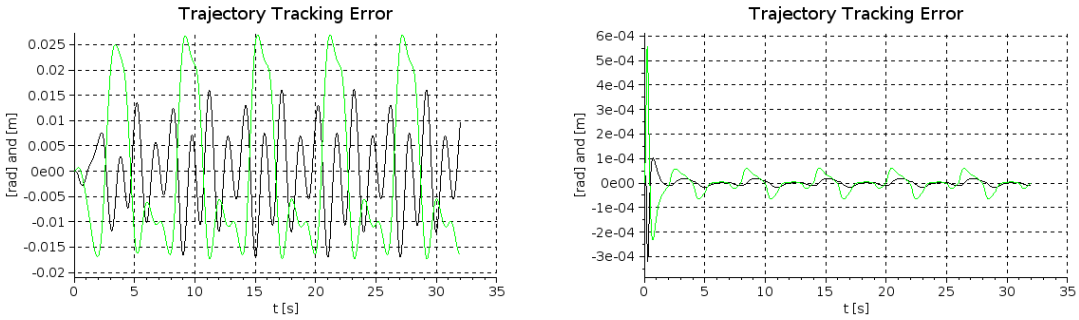

Figure 3.29: Trajectory tracking error in the non-adaptive (at the LHS) and the adaptive (at theRHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green lines]

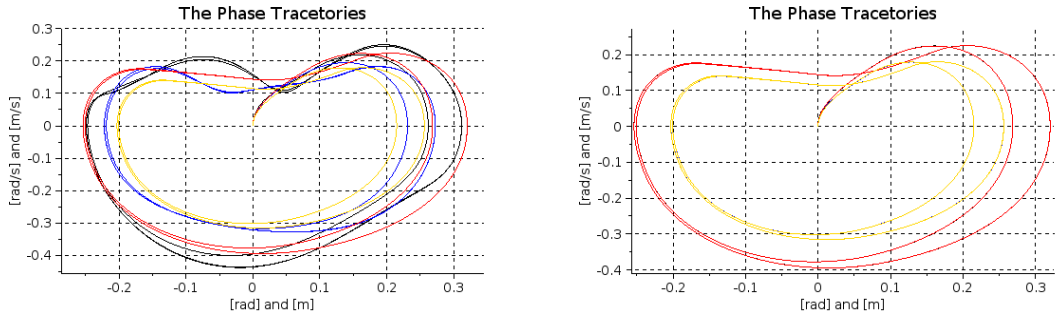

Figure 3.30: The phase trajectories for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [for q_2: black, q_3: blue, q_2^{Nom}: red, q_3^{Nom}: ocher lines]

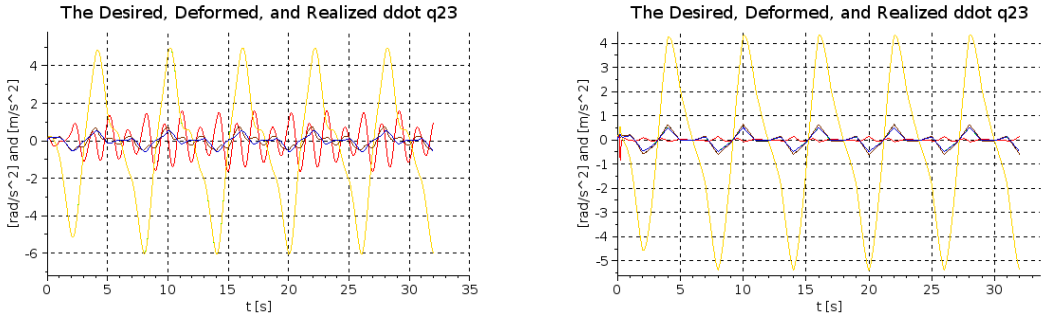

Figure 3.31: The *desired*, *adaptively deformed*, and the *realized* 2^{nd} time-derivatives for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines]

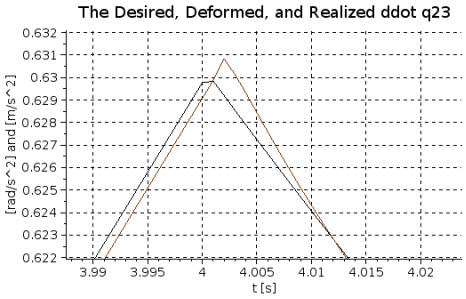

Figure 3.32: The *desired*, *adaptively deformed*, and the *realized* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines]

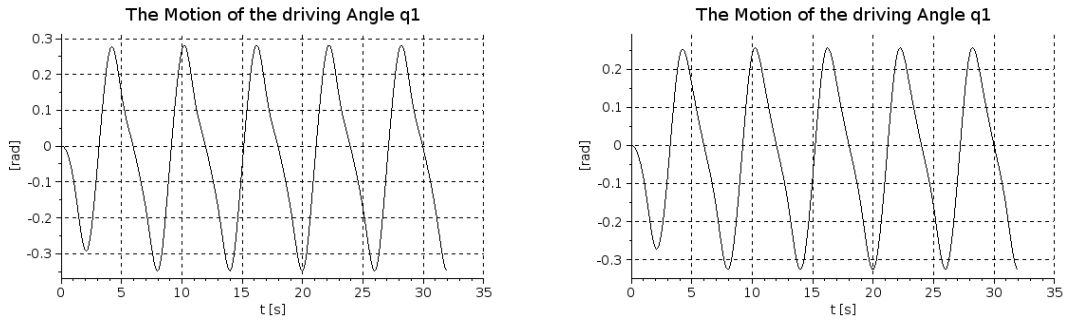

Figure 3.33: The trajectory of the driving arm q_1 for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases

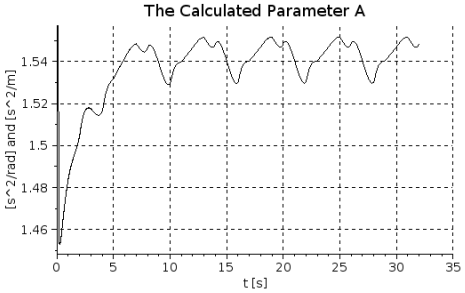

Figure 3.34: The calculated parameter A vs. time

3.4 Thesis Statement II.

I have introduced a new family of fixed point transformations that can be generated from sigmoid functions in similar manner as certain fuzzy aggregation, t-norm, and s-norm operators are produced by the use of appropriate generator functions. I have shown that by the use of this generation technique a pair of repulsive and attractive fixed points can be generated as in the case of the originally used RFPT. This new construction can be used for the adaptive control of SISO systems and also allows tuning only one of the control parameters.

3.4.1 Substatement I.

I have given an extension and its proof of the 'Sigmoid Generated Fixed Point Transformation' to MIMO systems. I have shown that: the only parameter of the controller A that must be set according to the dynamic properties of the system under control within a wide range can provide acceptable solution. The increase in its absolute value from a small initial one considerably can improve the tracking precision. However, the increase in this parameter quickly leads to oscillations in the iterative adaptive control.

3.4.2 Substatement II.

In the present statement I have given further improvements in order to enhance the precision of the suggested adaptive control scheme by replacing the parameter tuning with simple calculation. The proposed estimation and its digital realization is considerably simpler than the previously used.

Related publications: [A. 5], [A. 6], [A. 7], [A. 8]

Chapter 4

Advances in the Sigmoid Generated Fixed Point Transformation

4.1 Adaptive Control Using Improved Sigmoid Generated Fixed Point Transformation and Scheduling Strategy

The available model of the system under control normally is complicated and can be obtained at the costs of huge efforts ([63]). In certain cases typical "ranges of operation" can be identified that gives sufficient basis for the application of the idea of "Situational control" in the case of turbojet engines used in aviation [6]. However, when the characteristics of the system under control nonlinearly vary over a broad range of various conditions, the use of progressive and adaptive control approaches may be especially beneficial [64]. In a modern control solution "modeling", "control", and "diagnostic" elements are simultaneously present [65, 66]. When only approximate and incomplete system models are available the other widely acknowledged approaches in nonlinear control are the so-called optimal control and the Receding Horizon Controllers [67][68][69]. Typically the optimal controllers minimize some cost functional that is constructed of terms expressing various (often contradictory) requirements under the constraints that represent the dynamic properties of the system under control. The Receding Horizon controller is a special optimal controller that frequently redesigns the future horizon of the control so reducing the effects of modeling errors and unknown external disturbances. Optimal controllers use a cost function in order to ensure some trade–off between performance and accuracy. Common examples of application area are underactuated mechanical systems where it is impossible to drive them on an arbitrary 'trajectory' along by simultaneously precisely ensuring each state variables position in time. In order to distribute the tracking error between the state variable's a possible solution

is minimizing a cost function that is constructed as a sum of the errors. Due to a complicated mathematical framework of Lyapunov's Direct method it is far difficult to combine it with the optimal controllers. A possible solution by the combination of optimal controllers and adaptive controllers was suggested in [52] in the control of neuron models. With this in mind in this section I propose a strategy for replacing the optimal control by scheduling the control with time sharing in the adaptive control based on the improved SGFPT [A. 10].

4.1.1 New Function

The conditions for the stability for MIMO systems were proved in publication [A. 10]. This solution works in bounded region of attraction around $r_\star$ that formally cannot guarantee global stability. By shifting the function in the horizontal and vertical direction it was assumed that for the first element of the iteration x_0 there exists x_1 for which $g(x_0) - K = g(x - D)$. This is valid for most of the x_0 values but not for each of them. In order to overcome this limitation the following stretched function is proposed [A. 9]

$$F(x) = B\tanh(a(x+b)) + K \tag{4.1}$$

with $a, b > 0$. This way allows further a more precise positioning of the function in the vicinity of the solution of the control task [A. 9]. In the next sections the control design is presented for an underactuated Classical Mechanical System.

4.1.2 The Control Design for Underactuated Mechanical Systems

The phenomenon of swinging can be modeled by an *underactuated pendulum* according to Fig. 4.1: the internal degree of freedom of the human body is represented by a rotary axle $q_2\,[rad]$ with its own driving torque $Q_2\,[N \cdot m]$. The upper axle of the swing q_1 does not have any driving torque. To exemplify the applicability of the improved Sigmoid Generated Fixed Point Transformation with combination of optimal control an underactuated pendulum model serves as a benchmark problem. The dynamic model is given by Eqs. (4.2) and (4.3):

$$\begin{bmatrix} m_1L_1^2 + m_2L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos q_2 & m_2L_2^2 + m_2L_1L_2\cos q_2 \\ m_2L_2^2 + m_2L_1L_2\cos q_2 & m_2L_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + h = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{4.2}$$

$$h = \begin{bmatrix} -2m_2L_1L_2\sin q_2\dot{q}_1\dot{q}_2 - m_2L_1L_2\sin q_2\dot{q}_2^2 + [m_1+m_2]gL_1\sin q_1 + m_2L_2g\sin[q_1+q_2] \\ m_2L_1L_2\sin q_2\dot{q}_1^2 + m_2gL_2\sin[q_1+q_2] \end{bmatrix} \tag{4.3}$$

where q_1 and q_2 are rotary angles, Q_1 and Q_2 are joint torque values. In the possession of a single driving torque Q_1 *it is impossible to simultaneously precisely track nominal trajectories prescribed for both axles* as $q_1^N(t)$ and $q_2^N 1(t)$. The errors of tracking the components must be distributed between q_1 and q_2. Instead of using any cost function this distribution is realized by time-sharing: for a while the motion of q_1 is controlled while q_2 is let to move "as it wants", and in the other time-slot q_2 is controlled and during this session q_1 can move without direct control. The error distribution can be manipulated by properly modifying the time-slots [A. 9].

Further problem is that only an approximately known model is available for the control design (see Table 4.2).

Parameter	**Approximate value**	**Exact value**
$m_1\,[kg]$	2.2	2
$m_2\,[kg]$	0.8	1
$L_1\,[m]$	1	1
$L_2\,[m]$	2	2
$g\,[\frac{m}{s^2}]$	10	9.81

Table 4.1: The parameters of the "Approximate Model" and the actually controlled system's exact model

The suggested control method at first transforms the control task into a fixed point problem then solves it via iteration: in each time-step one iteration can be done during the digital control.

4.1.2.1 Realization of the Suggested Control Method using Stretched Sigmoid Function

For the control a novel fixed point transformation (SGFPT) is used that has been published in [A. 5], the structure of the simulation is given in Fig. 4.2. It is made by the use of the SCILAB-XCOS simulator.

The controller's parameters are given in the "context box" of the graphical simulation. The parameter `Adaptive` must be set to 1 for adaptive, and 0 for non-adaptive control.

4.1.2.2 Simulation Results

For the simulation the *exact model parameters* and *approximate model parameters* are set according to 4.2. The *kinematic trajectory tracking* is assumed to be $\ddot{q}_i^{Des} = \ddot{q}_i^{Nom} +$

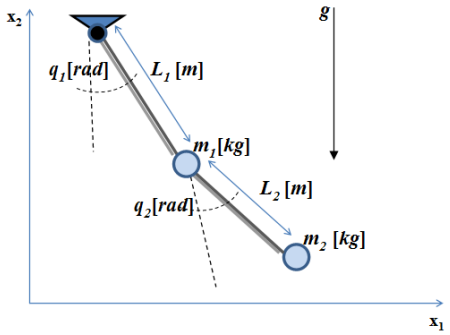

Figure 4.1: Example of the *"swinging paradigm"*: in the *underactuated system* axle q_1 has no driving torque, i.e. the appropriate generalized force $Q_1\,[N] \equiv 0$. The driving torque of axle q_2 i.e. $Q_2\,[N]$ is used for the realization of a compromise in approximately tracking a nominal trajectory $q_1^N(t) \neq 0$ and $q_2^N(t) \equiv 0$. (This latter restriction is introduced for saving the body of the swinging child.)

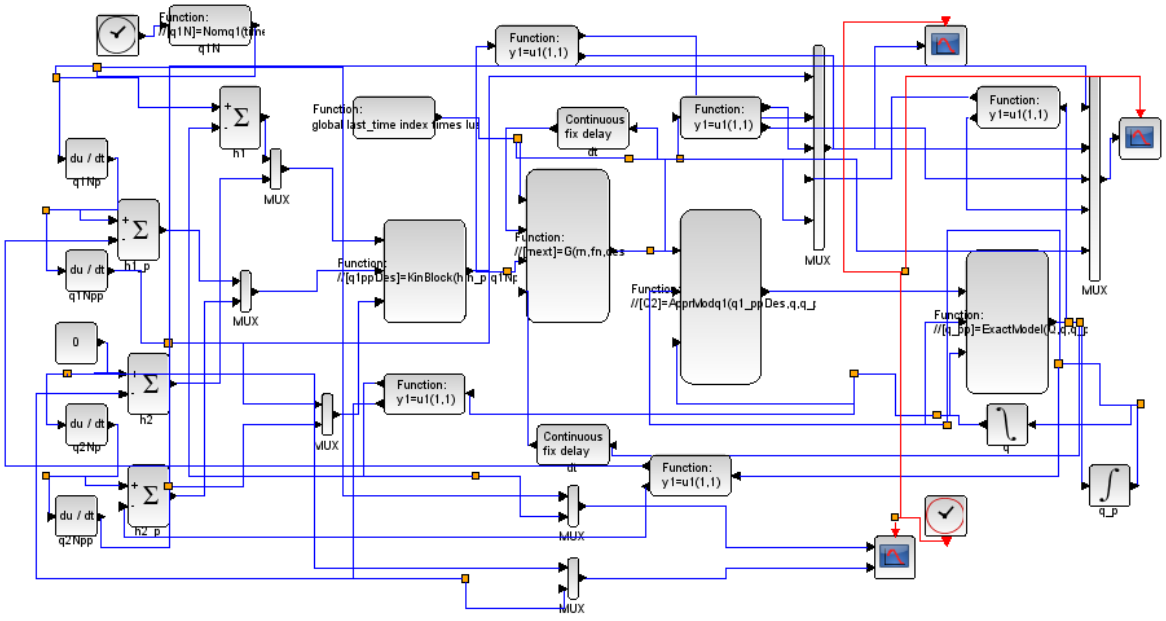

Figure 4.2: The structure of the controller and the simulation

$3\Lambda^2(q_i^{Nom} - q_i) + 3\Lambda(\dot{q}_i^{Nom} - \dot{q}_i) + \Lambda^3 \int_0^t \left(q_i^{Nom}(\tau) - q_i(\tau)\right) \mathrm{d}\tau$ with a time constant $\Lambda = 6\,[s^{-1}]$.

The results on the trajectory tracking can be observed in Fig. 4.4 for the adaptive case while Fig. 4.3 shows the results for the non-adaptive one.

In Figs. 4.5 the detail of the operation of the adaptivity due to the adaptive deformation of the input can be observed.

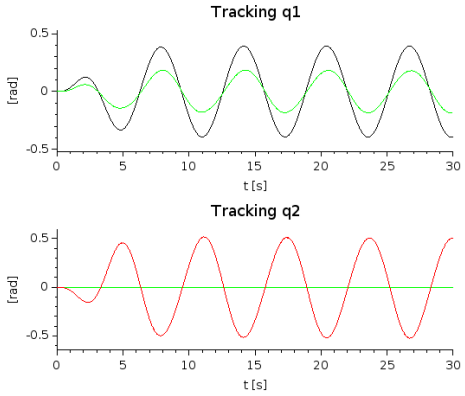

Figure 4.3: Non-adaptive trajectory tracking; top: $q_1^N(t)$: black line, $q_1(t)$: green line; bottom: $q_2^N(t)$: green line, $q_2(t)$: red line

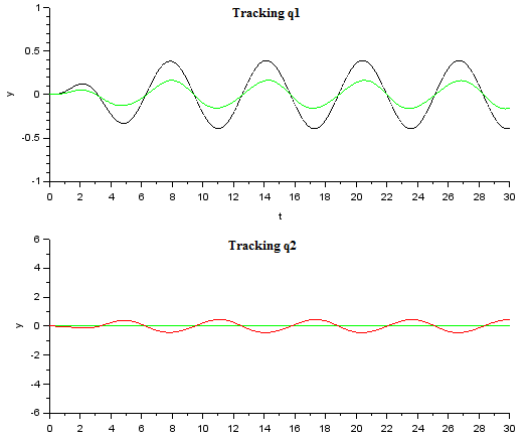

Figure 4.4: Adaptive trajectory tracking; top: $q_1^N(t)$: black line, $q_1(t)$: green line; bottom: $q_2^N(t)$: green line, $q_2(t)$: red line

Figure 4.5 reveals that in the non-adaptive phase the *nominal*, the *kinematically designed "desired"* values (that contain the PID error-corrections of the nominal trajectory) and the *realized* second derivatives seriously differ to each other. (In the lack of adaptivity the "desired" and the "deformed" values are exactly identical.) In the adaptive case the "desired" and the "deformed" values considerably differ from each other, but the "realized" value has a fast convergence to the "desired" one, that is the kinematically prescribed tracking is realized in the appropriate time-slot. The same holds for the control of axle q_2 (Fig. 4.6).

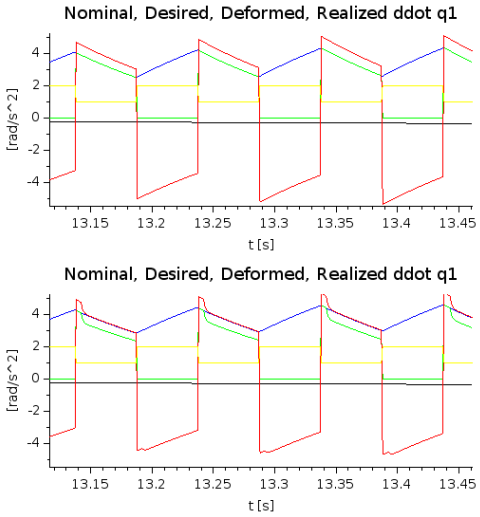

Figure 4.5: Time-dependence of $\ddot{q}_1$ (zoomed-in excerpts): non-adaptive control: top, adaptive control bottom (color codes: black line: $\ddot{q}_1^N(t)$ nominal, blue line: $\ddot{q}_1^{Des}$ kinematically prescribed "desired", green line: $\ddot{q}_1^{Def}$ adaptively deformed, red line: $\ddot{q}_1$ realized (simulated), yellow line: the timer: for 1 q_1 is under control, for 2 q_2 is under control)

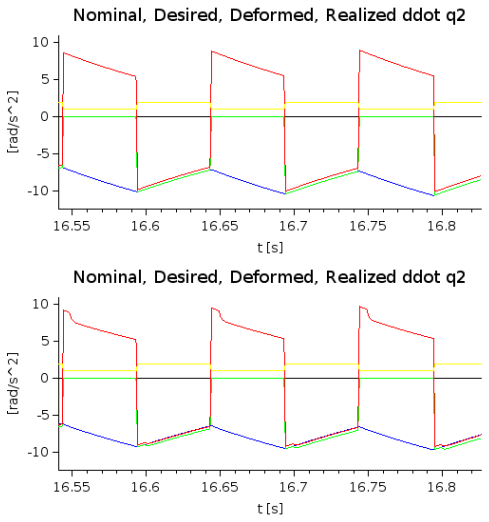

Figure 4.6: Time-dependence of $\ddot{q}_2$ (zoomed-in excerpts): non-adaptive control: top, adaptive control bottom (color codes: black line: $\ddot{q}_2^N(t)$ nominal, blue line: $\ddot{q}_2^{Des}$ kinematically prescribed "desired", green line: $\ddot{q}_2^{Def}$ adaptively deformed, red line: $\ddot{q}_2$ realized (simulated), yellow line: the timer: for 1 q_1 is under control, for 2 q_2 is under control)

4.2 Novel Type of Function

The results provided in 4.1 indicated further investigations on the class of applicable functions. I have found that instead of Eq. (4.1) the following new type of function belonging to

this family can be applied [A. 10]:

$$F(x) = \text{atanh}(\tanh(x + D)/2). \tag{4.4}$$

The following section presents application example and numerical results that support the advantages of the control scheme using Eq. (4.4) in the fixed point transformation.

4.2.1 Validation of Practical Applicability Through the Adaptive Control of Kapitza's Pendulum System

In this example a simplified Kapitza pendulum system serves as a benchmark problem. Stabilization problems of oscillating motions in pendular systems are well studied for horizontal movement of the pivot [70] [71]. The inverted pendulum on a pivot point that vibrates in a vertical direction, upside- down is the so-called Kapitza's pendulum that is used as the example of parametric oscillator in nonlinear control theory [72]. Under a high-frequency vertical excitation of the pivot point, a complex periodic motion of the pendulum can be observed in the vicinity of the upright position referred to as induced or vibrational stability. Models of pendulum with vertical vibration of the pivot and their properties have been widely investigated [73] [74] [75] [76] [77] [78]. Due to its unique features it is also used as a benchmark system for many other technological, physical and natural phenomena including biological processes, vibrational technologies, anti-gravity problems, etc. The scheme of the simplified model under consideration can be seen in Fig. 4.7 [A. 10]

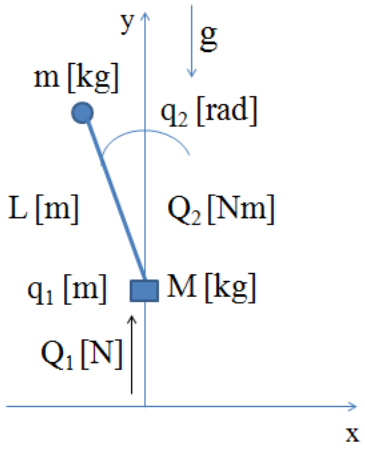

Figure 4.7: The scheme of the upside-down pendulum system

In the case of the *underactuated model* $Q_2 \equiv 0$ and we can prescribe the *"nominal trajectories"* for q_2 and adjust the driving force value Q_1 by using the following equation of motion:

$$\begin{bmatrix} (M+m) & -mL\sin q_2 \\ -mL\sin q_2 & mL^2 \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \end{bmatrix} + \begin{bmatrix} -mL\cos q_2 \dot{q_2}^2 + (m+M)g \\ -mgL\sin q_2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{4.5}$$

For defining the *desired* $\ddot{q_1}^{Des}$ value we can write [A. 10]:

$$\ddot{q_1}^{Des} = \frac{mL^2\ddot{q_2} - mgL\sin q_2}{mL\sin q_2} = \frac{L}{\sin q_2}\ddot{q_2}^{Des} - g \tag{4.6}$$

In order to maintain the dynamic singularity that has occured in Eq. (4.6) I have introduced the following kinematic strategy;

$$\ddot{q_1}^{Des} = L\frac{|q_2|}{\sin q_2}\ddot{q_2}^{Des} - g. \tag{4.7}$$

with the conditions below:

```
sq2=sin(q2)
  q1_ppDes_max=La*q2_ppDes/sin(error_limit);
  q1_ppDes_min=-q1_ppDes_max;
  if abs(sq2)>error_limit then
    q1_ppDes=La*q2_ppDes/sq2;
  else
    if sq2<=0 then
      q1_ppDes=q1_ppDes_min;
    else
      q1_ppDes=q1_ppDes_max;
```

4.2.1.1 Results of Numerical Simulations

For the simulation the *exact model parameters* are set in $m = 2\,[kg]$, $M = 1\,[kg]$, where m represents the mass attached the end of the pendulum and M is the mass at the bottom of the pivot. The length of the beam is set in $L = 1\,[m]$ and it's mass is neglected. The following *approximate model parameters* are used: $\tilde{m} = 2.2\,[kg]$, $\tilde{M} = 1.5\,[kg]$, $\tilde{L} = 0.8\,[m]$, and $g = 9.81\,[m/s^2]$. The PID-type *kinematic trajectory tracking* is assumed to be $\ddot{q}_2^{Des} = \ddot{q}_2^{Nom} + 3\Lambda^2(q_2^{Nom} - q_2) + 3\Lambda(\dot{q}_2 i^{Nom} - \dot{q}_2) + \Lambda^3 \int_0^t \left(q_2^{Nom}(\tau) - q_2(\tau)\right) \mathrm{d}\tau$ with a time constant $\Lambda = 3\,[s^{-1}]$. The simulation is carried out by using SCILAB-XCOS simulator. The structure of the simulation is depicted in Fig.(4.8). In the *adaptive controller* the controller cycle time is $\delta t = 10^{-4}\,[s]$, the value of A is set -10 while $D = -0.3$. The results on the trajectory tracking and the trajectory tracking errors can be observed in Figs. 4.9, 4.10 and

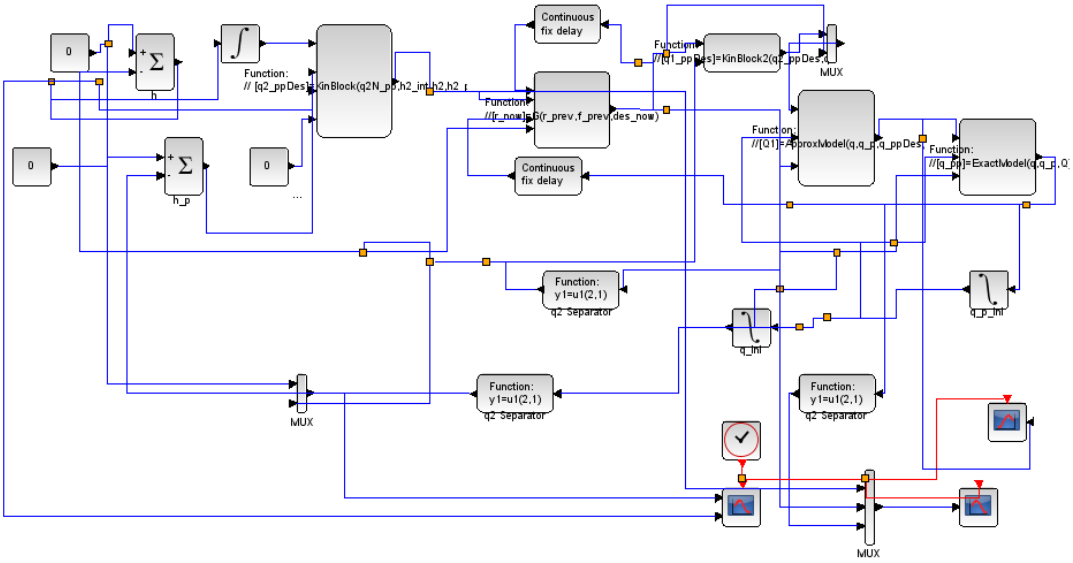

Figure 4.8: The structure of the controller and the simulation

Fig. 4.11 for both the adaptive and non-adaptive cases. It can be observed, that the adaptive strategy results in improvement on the precision compared to the non-adaptive case. In Figures 4.12 and 4.13 the operation of the adaptivity according to the adaptive deformation of the input can be seen. The desired $\ddot{q}_2^{Des}$ which is represented with black line and $\ddot{q}_2$ shown with green line are covering each other. The values for Q_1 are displayed in Figs. 4.14 and 4.15. The results of the numerical simulation demonstrate, that we have achieved accurate trajectory tracking with the new type of $sigmoid$ function applied in the proposed control scheme.

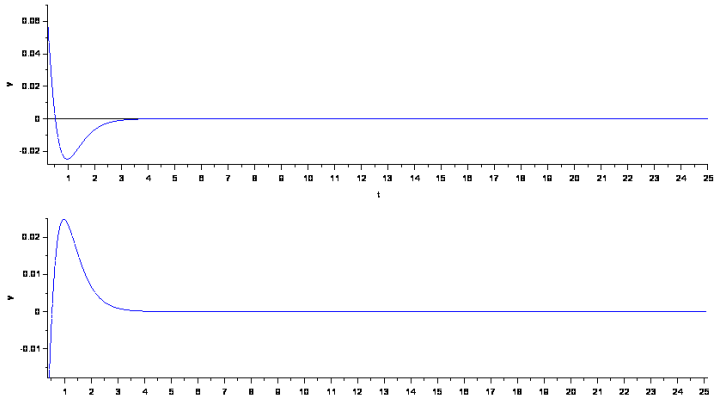

Figure 4.9: Trajectory tracking and tracking error in the the adaptive case

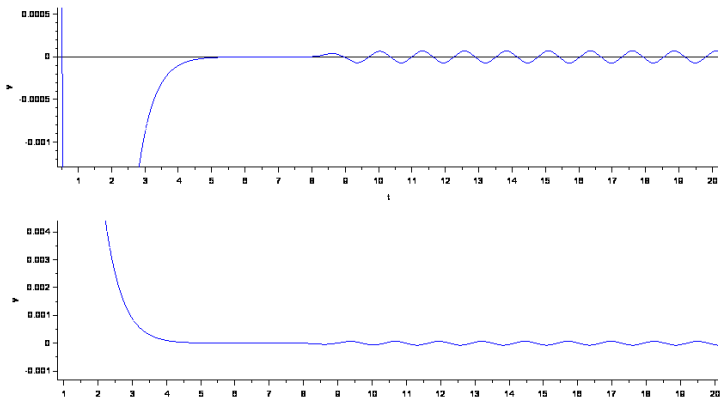

Figure 4.10: Trajectory tracking and tracking error in the the adaptive cases -zoomed

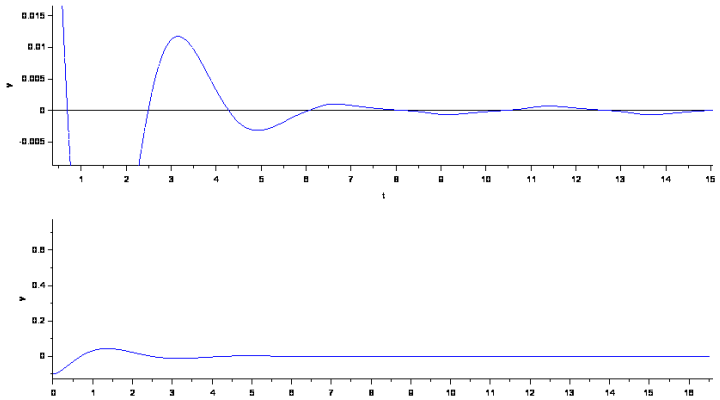

Figure 4.11: Trajectory tracking and tracking error in the the non-adaptive cases -zoomed

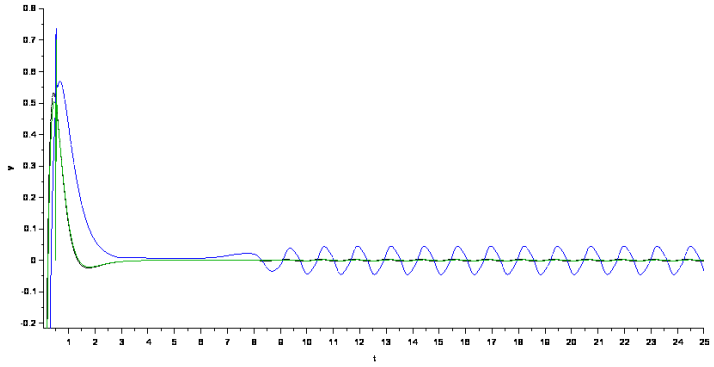

Figure 4.12: Time-dependence of $\ddot{q}_2$ in case of adaptive control (color codes: black line:$\ddot{q}_2^{Des}$ kinematically prescribed "desired", blue line: $\ddot{q}_2^{Def}$ adaptively deformed, green line: $\ddot{q}_2$ realized (simulated)

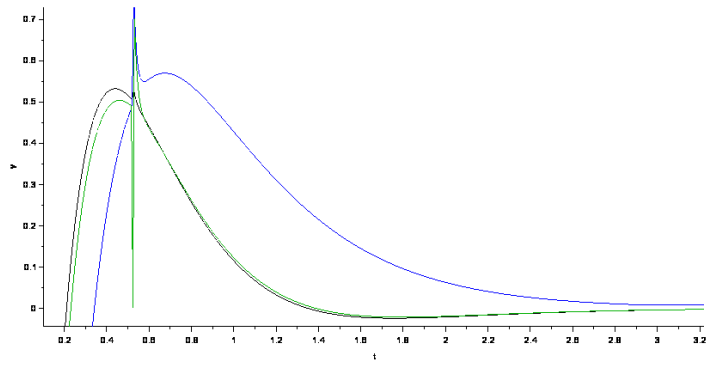

Figure 4.13: Time-dependence of $\ddot{q}_2$ in case of adaptive control -zoomed (color codes: black line:$\ddot{q}_2^{Des}$ kinematically prescribed "desired", blue line: $\ddot{q}_2^{Def}$ adaptively deformed, green line: $\ddot{q}_2$ realized (simulated)

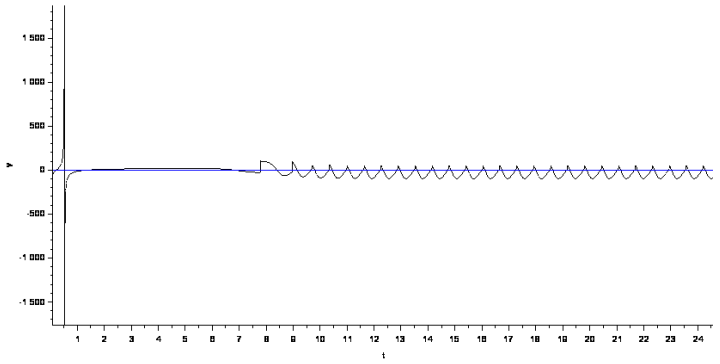

Figure 4.14: Q1 [N] vs. time in the adaptive case

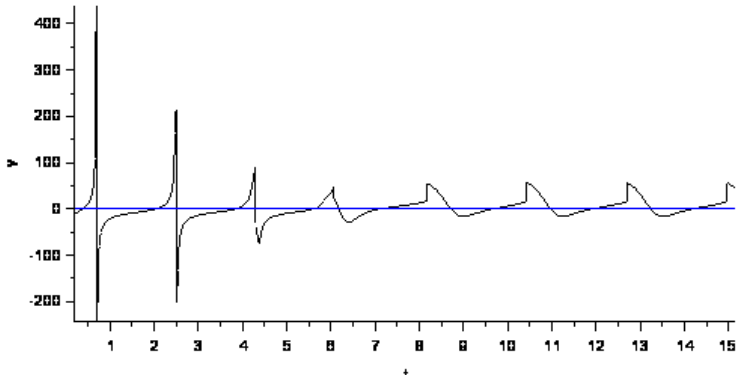

Figure 4.15: Q1 [N] vs. time in the non-adaptive case

4.3 Enhancement of the SGFPT Control Design by Soft Computing

The excellent properties of soft computing methods, such as fuzzy logic, neural networks and evolutionary computing provide a wide range of tools for addressing issues of establishing relationship between measurements, assigning error bars to predictions, integrating information from various sources with a varying degree of uncertainty, etc. [79] [80]. Recently, most of the deterministic model buildings are increasingly replaced by soft-computing methods [81]. Inspite of the classical hard-computing methods the intelligent methodologies are able to deal with imprecisions, uncertainties and partial truth by an efficient and robust way. Fuzzy logic is widely used for modeling complex and ill-defined systems. The core concept relies on the application of linguistic variables, see [82]. In this section a possible combination of the fuzzy modeling and the SGFPT control strategy is shown [A. 11]. The inverted pendulum system serves as a nonlinear paradigm.

4.3.1 The System under Consideration

The proposed strategy is demonstrated by the control of the following inverted pendulum system. The generalized coordinates are $q_1\,[rad]$, $q_2\,[rad]$ and the generalized forces are torque signals: $Q_1\,[N\cdot m]$, $Q_2\,[N\cdot m]$. The equations of motion are given in (4.8).

$$\begin{pmatrix} mL^2 & 0 \\ 0 & mL^2\sin^2 q_1 \end{pmatrix}\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix}+\begin{pmatrix} -mL^2\sin q_1\cos q_1\dot{q}_2^2+mgL\sin q_1 \\ 2mL^2\sin q_1\cos q_1\dot{q}_1\dot{q}_2 \end{pmatrix}=\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \tag{4.8a}$$

The system parameters are collected in Table 4.2.

Parameter	**Exact Value**	**Approximate Value**
mass m	$0.5\,kg$	$1\,kg$
lenght L	$1.5\,m$	$1\,m$
gravitational acceleration g	$9.81\,\frac{m}{s^2}$	$10\,\frac{m}{s^2}$

Table 4.2: The system model and its parameters

4.3.2 The Control Strategy

The fixed point transformation is carried out as described previously. The iteration generates sequences as $\{r(i) \in \mathbb{R}^n | i = 0, 1, \ldots\}$ by using the direction of the *response error* in the i^{th} step of the iteration as

$$h(i) \stackrel{def}{=} f(r(i)) - r^{Des}, e(i) \stackrel{def}{=} \frac{\mathfrak{A}h(i)}{\|\mathfrak{A}h(i)\|}, \tag{4.9}$$

where $\|h\|$ is the Frobenius norm:

$$\begin{aligned} r(i+1) = \tilde{G}(r(i)) \stackrel{def}{=} \\ \stackrel{def}{=} [F(A\|\mathfrak{A}h(i)\| + x_*) - x_*]\, e(i) + r(i) \ . \end{aligned} \tag{4.10}$$

Afterwards, the solution of the control task is the fixed point of $\tilde{G}(r)$ when $f(r_\star) - r^{Des} \equiv h(i) = 0$ then $r(i+1) = r(i) = r_\star$.

In (4.9) and (4.10) $\mathfrak{A}$ is a diagonal matrix with positive main diagonals that can be tuned to improve the convergence properties of the controller. (In the original approach it was the unit matrix.) Its matrix elements can be tuned by observing little fluctuations in the convergence of the adaptive signal when these main diagonals are too big. These fluctuations are revealed as a negative content in a forgetting buffer as it was done in [43].

The *constant control parameters* can be found in Table 4.3. The simulations are carried out by using the package *"Julia"* with a sequential code using Euler integration method with a fixed step length of $10^{-4}\, s$.

Parameter	**Value**
Λ	$4\, s^{-1}$
D	0.3
δt time delay in learning	$10^{-3}\, s$

Table 4.3: Setting of the constant control parameters

A PID-type relaxation can be prescribed for the tracking error of q. Let $\mathbb{R} \ni \Lambda > 0$, and let

$$e(t) \stackrel{def}{=} q^N(t) - q(t) \ , \tag{4.11a}$$

$$e_{int}(t) = \int_{t_0}^{t} \left(q^N(\xi) - q(\xi)\right) \mathrm{d}\xi \ , \tag{4.11b}$$

$$\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^2 e(t) = 0 \Rightarrow \tag{4.11c}$$

$$\ddot{q}^{Des} = \ddot{q}^N + \Lambda^3 e_{int}(t) + 3\Lambda^2 e(t) + 3\Lambda \dot{e}(t) \ . \tag{4.11d}$$

The adaptive controller at first deforms $\ddot{q}^{Des}$ into $\ddot{q}^{Def}$, and uses the approximate dynamic model for the calculation of the control forces for this deformed value. According to (4.8), for this excitation the controlled system responses with the *"Realized"* response $\ddot{q}$. On the sequel function $\ddot{q} = f\left(\ddot{q}^{Def}\right)$ is referred to as the *"response function"* of the system under control. The adaptive deformation is based on the measurements of these responses. The controller used the function 4.4 for making the adaptive deformation.

4.3.3 Results for the Affine Model

At first, let's investigate the affine model. The simulation results can be seen in Figs. 4.16-4.19. The adaptive strategy is compared with a non-adaptive one. It is evident that the modeling errors cause errors in the computed torque signals and corrupt the precision of trajectory tracking [A. 11].

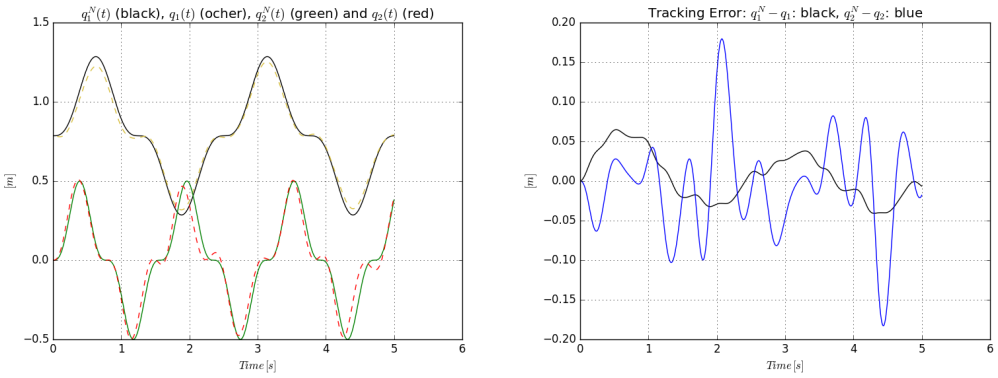

Figure 4.16: Trajectory tracking and tracking error of the non-adaptive controller for the *"affine model"*

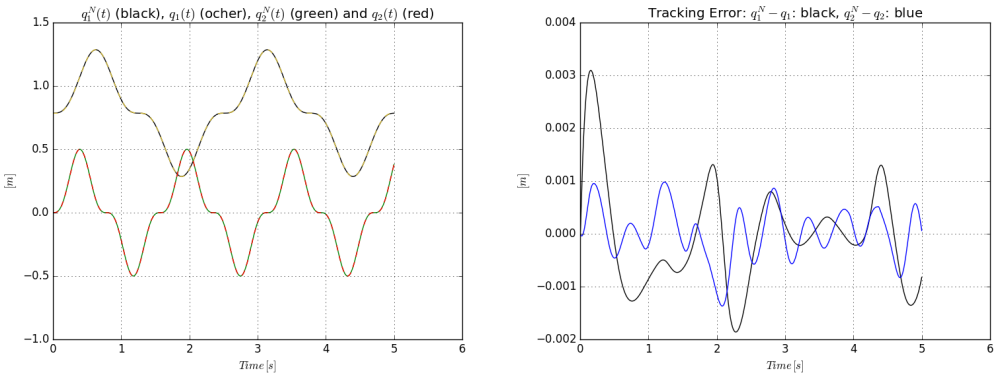

Figure 4.17: Trajectory tracking and tracking error of the adaptive controller for the *"affine model"*

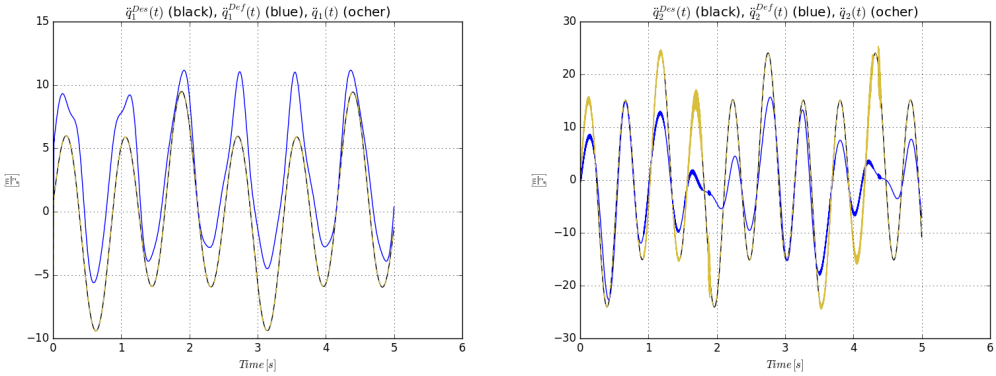

Figure 4.18: The $\ddot{q}$ values of the adaptive controller for the *"affine model"*

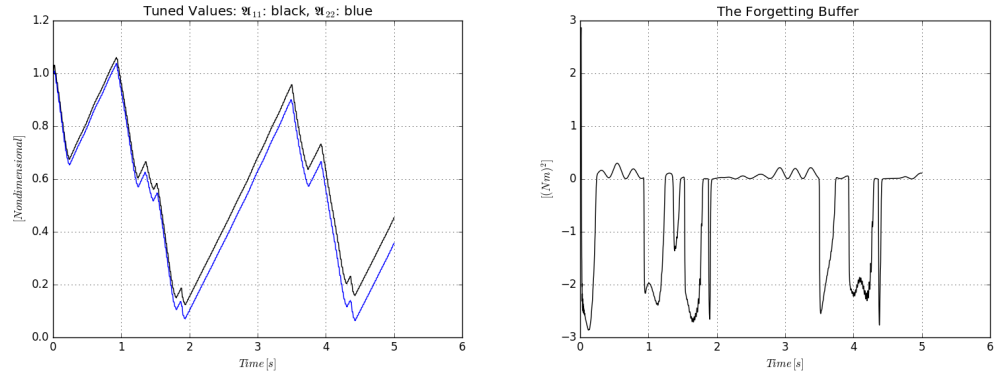

Figure 4.19: The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"affine model"*

4.3.4 Results for the Soft Computing-based Model

In this case of the Soft Compting-based model the $\sin(x)$ and the $\cos(x)$ functions are approximated by fuzzy rules as $\mu_s(x)$ and $\mu_c(x)$ over a bounded region [A. 11], according to Fig. 4.20.

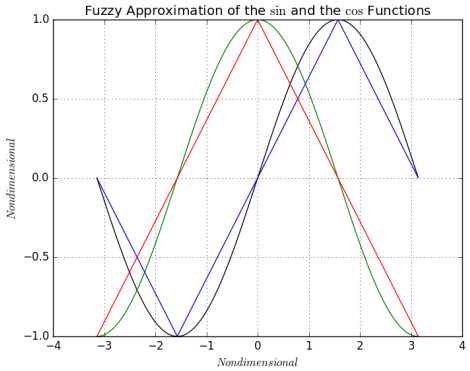

Figure 4.20: Functions $\sin(x)$ (black), $\mu_s(x)$ (blue), $\cos(x)$ (green), and $\mu_c(x)$ (red)

In the *approximate dynamic model* the additional terms are constants just as in the case of the "affine model", but the function $\sin(x)$ in the inertia matrix is approximated:

```
function ApprMod(q,q_p,q_ppDes)
  global ma
  global La
  global ga
  H=zeros(2,2)
  sq1=mus(q[1,1])
  H[1,1]=ma*La^2;
  H[2,2]=ma*La^2*sq1^2;
  h=[1.0;1.0];
  return H*q_ppDes+h;
end
```

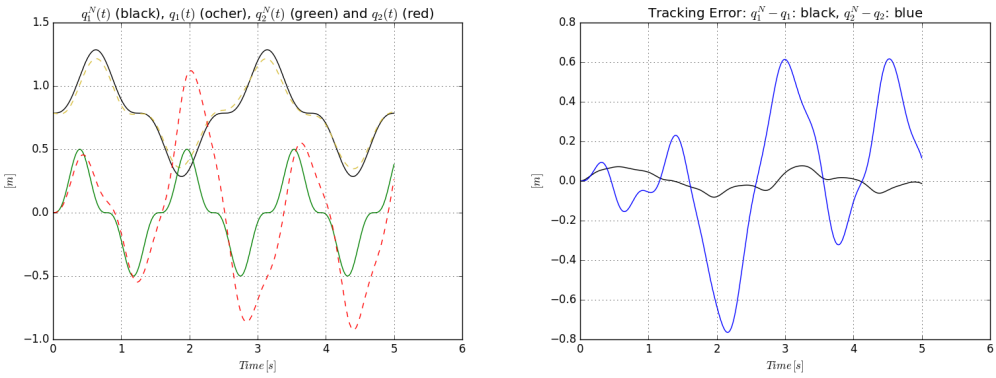

Figure 4.21: Trajectory tracking and tracking error of the non-adaptive controller for the *"soft computing-based model"*

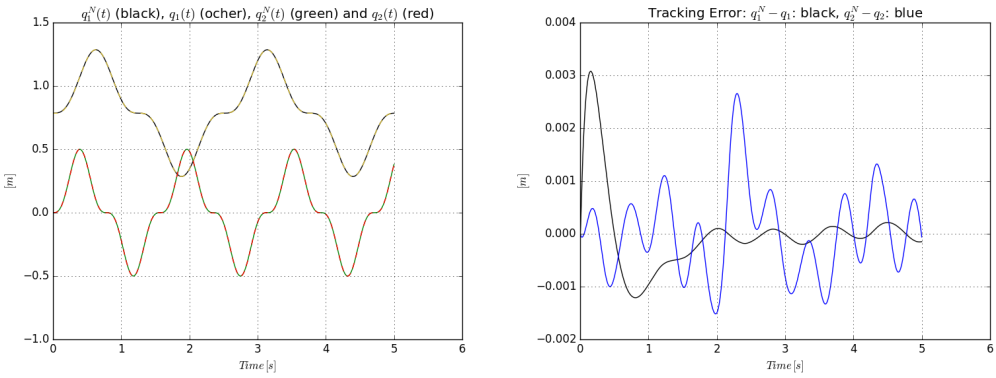

Figure 4.22: Trajectory tracking and tracking error of the adaptive controller for the *"soft computing-based model"*

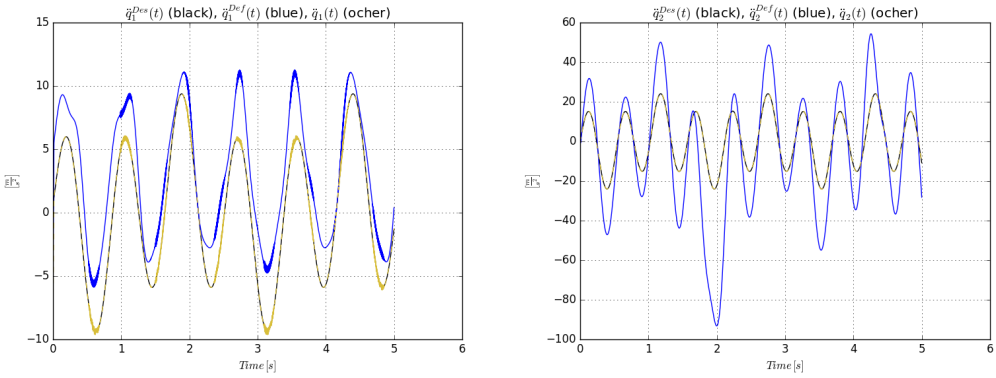

Figure 4.23: The $\ddot{q}$ values of the adaptive controller for the *"soft computing-based model"*

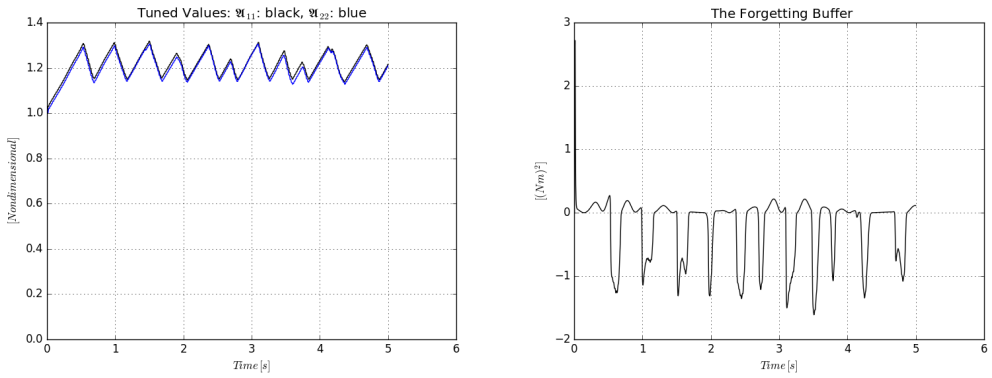

Figure 4.24: The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"soft computing-based model"*

4.3.5 Results for the Fully Soft Computing-based Model

In the *approximate dynamic model* in each term the functions $\sin(x)$ and $\cos(x)$ are approximated by the fuzzy approximations $\mu_s(x)$ and $\mu_c(x)$ as in [A. 11],

```
function ApprMod(q,q_p,q_ppDes)
 global ma
 global La
 global ga
 H=zeros(2,2)
 sq1=mus(q[1,1])
 cq1=muc(q[1,1])
 H[1,1]=ma*La^2;
 H[2,2]=ma*La^2*sq1^2;
 h=[1.0;1.0];
 h[1,1]=-ma*La^2*sq1*cq1*q_p[2,1]^2+
 +ma*ga*La*sq1;
 h[2,1]=2*ma*La^2*sq1*cq1*q_p[1,1]*q_p[2,1];
 return H*q_ppDes+h;
end
```

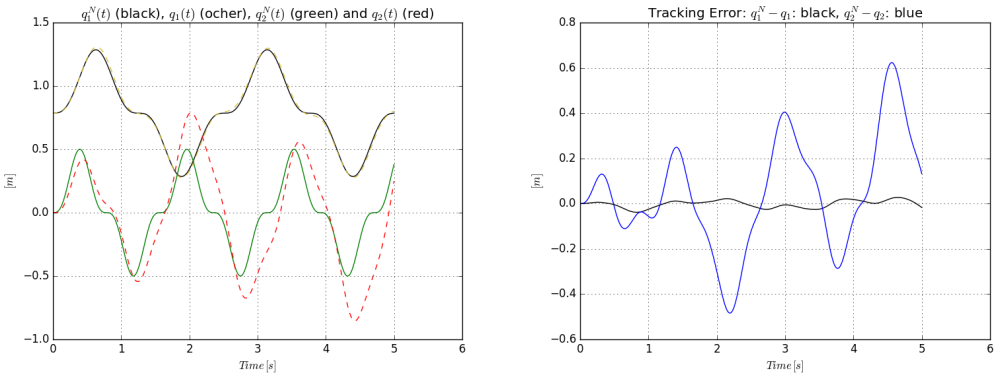

Figure 4.25: Trajectory tracking and tracking errorof the non-adaptive controller for the *"soft computing-based model"*

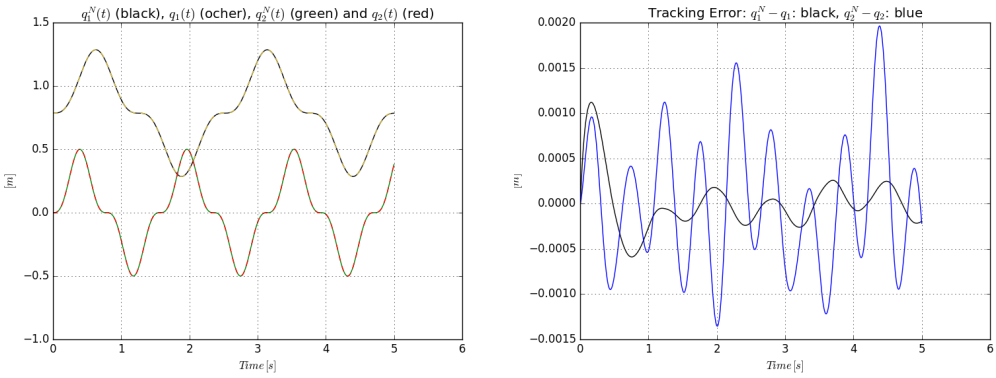

Figure 4.26: Trajectory tracking and tracking error of the adaptive controller for the *"fully soft computing-based model"*

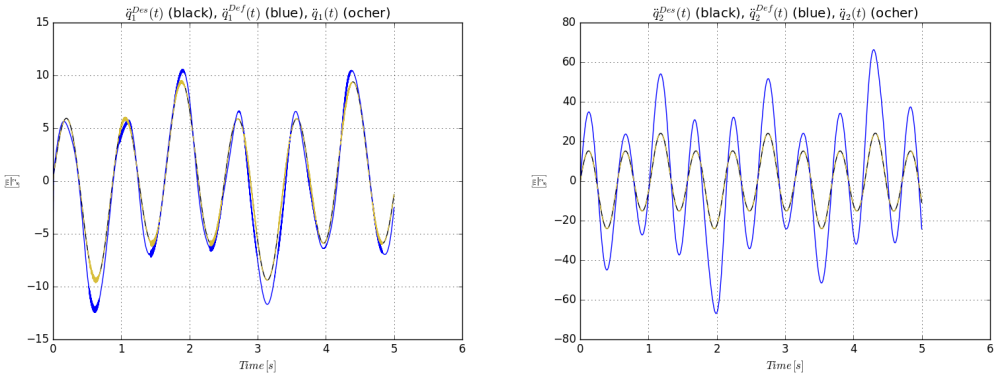

Figure 4.27: The $\ddot{q}$ values of the adaptive controller for the *"fully soft computing-based model"*

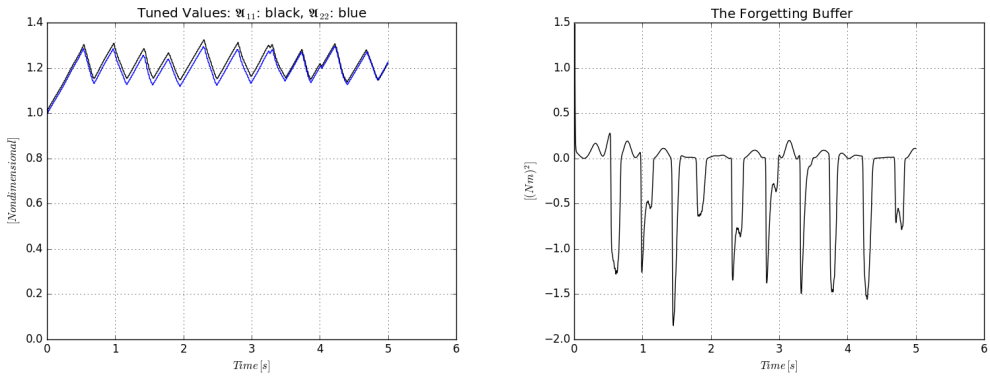

Figure 4.28: The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"fully soft computing-based model"*

4.4 Thesis Statement III.

In this Thesis I have introduced the new Stretched Sigmoid function for the SGFPT-based control design. In this improved technique the sigmoid is stretched instead of simple shifting. The main advantage is that it allows excellent positioning in the vicinity of the solution of the control task. The new design has been introduced by the combination of optimal controllers and adaptive controllers that allows overcoming the difficulties of controlling underactuated mechanical systems.

4.4.1 Substatement I.

I have introduced a new function of this family. The applicability and effectiveness of the proposed control method using this novel type of function have been confirmed by the adaptive control of the inverted pendulum with vertical vibration of the pivot.

4.4.2 Substatement II.

I have shown that the Improved SGFPT type control design can be supported by soft computing techniques. The approximations of trigonometric functions of the dynamic model has been realized with fuzzy rules. I have proved via comparative simulation investigations of an "affine" and a "soft computing-based" model that this new construction is able to deal with imprecisions, uncertainties, etc. by an efficient and robust way.

Related publications: [A. 9], [A. 10], [A. 11]

Chapter 5

Improved Denoising in the Wavelet Domain

Most of the applications of adaptive signal processing and data representation, etc., require the development of highly efficient data processing techniques. The classical approaches, such as linear filtering, can smooth the corrupted signal, but with weak feature localization and incomplete noise suppression. These are especially important in applications that require online responses. Nonlinear filters have been proposed to overcome these limitations. Among the fundamental methods of signal processing, wavelet-based noise reduction has been successfully applied to filter data. This technique is so advantageous due to providing information at a level of detail, that is not available with the Fourier methods [83]. The discrete wavelet transform (DWT) analyses the signal at different frequency scales with different resolutions by reducing the signal into approximate and detail coefficients [83] [84]. For removing noise, wavelet shrinkage employs nonlinear soft thresholding functions in the wavelet domain. An ample number of papers have been published on the possible extensions and areas of applications of the discrete wavelet transform, for instance [85].The fast signal transform algorithms provide significant reduction in the computation time. The effectiveness of wavelet shrinkage relies on that the wavelet transforms the additive white noise into white noise in the coefficient domain. Thus, fewer coefficients represent the signal which allows proper separation of the noise. Further, the wavelet-based technique requires only less assumption about the properties of the signal. These advantageous properties and possible applications have been investigated in [A. 12][A. 13]. Otherwise, one must be cautious in applying this technique, because by using higher decomposition levels, the signal loses more of its important features which may degrade the result significantly. Furthermore, the specific choice of the wavelet function, decomposition level, and thresholding rule allows to construct a large number of shrinkage procedures. Advanced concepts on the thresh-

old function have been introduced in [86].The fundamental criteria behind the construction of shrinkage procedures are robustness, adaptivity to the data, information of interest, and continuous operation. In critical situations or when failures occur in the processing system anytime techniques can be applied to carry on continuous operation [3]. In such situations the required reaction time is shortened and the processing phase must be completed in time even in the absence of all the required data or computational power. The principle of anytime design lies in that if there is a loss of some data or resources, the current operation can be continued based on algorithms providing short response time [87][88]. On the cost of accuracy optimal overall performance can be maintained.

5.1 Wavelet Shrinkage

The two main approaches of the denoising task are the processing in the time or space domain and the processing in the transform domain [89]. The methods performed in the transform-domain assume that the original signal can be well approximated by a linear combination of some basis functions. The wavelet transform preserves the true signal in few high-magnitude wavelet coefficients while others are associated with noise. Formally, let us consider the classical problem of noise removal: $y_i = f(t) + \epsilon\,, i = 1, \ldots, s$ where y_i denotes the observed noisy data and ϵ represents the random noise, which is an independent and identically distributed (iid) process, and (t) stands for time. Let f denote the unknown function. The sampling points are equally spaced $s = 2^n$ in order to allow to perform the discrete wavelet transform (DWT). The issue is to estimate f on $y_i = [y_1, \ldots, y_s]$ with minimum risk in least squares sense, The first step of wavelet shrinkage is the decomposition of y_i as follows; $y_{ij} = \omega_{ij}\epsilon_i$, where ω_{ij} are the wavelet (detail and approximate) coefficients on j^{th} scale. The general idea behind wavelet shrinkage is to replace the coefficients with small magnitude to zero (hard thresholding) or set their value to the threshold level. After, the reconstruction is carried out by performing the inverse discrete wavelet transform (IDWT). Generally, the shrinkage methods construct nonlinear threshold functions based on some statistical considerations. For instance, the smoothness-adaptive method (SureShrink) [90] is proposed to threshold each dyadic resolution level using the principle of Stein's Unbiased Estimate of Risk [91] , while the universal bound thresholding rule provides results with low computational complexity. The rule of the latter is defined as follows [90],

$$\nu_1 = \sigma_{MAD}\sqrt{2}\log s_j \tag{5.1}$$

where $\sigma_{MAD} = \frac{median(\omega_j)}{0.6745}$ denotes the absolute median deviation. The SureShrink procedure, which removes noise by thresholding the empirical wavelet coefficients is the following.

Suppose a vector of the squared wavelet coefficients in increasing order, $\Omega = [\omega_1, \ldots, \omega_k]$. The risk of these coefficients are calculated as $r = [r_1, \ldots, r_k] = \frac{\left|s-2i+(s-i)\omega_i+\sum_{i=1}^{k}\omega_1^2\right|}{s}$ and the threshold equals

$$\nu_2 = \sigma\sqrt{\omega} \tag{5.2}$$

where ω is the smallest element of the risk vector and σ denotes the standard deviation of the noisy signal.

The Heuristic Sure thresholding rule is a heuristic combination of the Sure method and the universal bound [90],

$$\nu_3 = \begin{cases} \nu_1, & \text{if } p \leq q \ . \\ min(\nu_1, \nu_2), & \text{otherwise.} \end{cases} \tag{5.3}$$

in which $p = \frac{m-k}{k}$, $q = (log_2 k)^{3/2}$ and $m = \sum_{i=1}^{k}\omega_i^2$. The principle of minimax rule is based on the estimator design used in statistics, the threshold is given by [90],

$$\nu_4 = \begin{cases} \nu_{MAD}(0.3936 + 0.1829s), & \text{if s>32.} \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

5.1.1 Fuzzy Supervisory System

Presently fuzzy systems are being applied to various process supervision tasks in an effective way [A. 14]. When failures appear, we often wish to provide continuous operation. To address this challenge, a fuzzy expert system has been improved for anytime signal auto-healing. Anytime mode of operation is able to cope with missing information [92][93]. The scheme of the supervisory system is depicted in Fig. (5.1). The inputs of the Mamdani-type fuzzy expert are created by the measurements of the noisy signal and the sine probe function. The first fuzzy module assigns fuzzy values to energy attributes and to the signal to noise ratio (SNR). Based on the rules taken from its database, the expert system evaluates the necessary wavelet resolution level. The second fuzzy module selects the most suitable thresholding rule from $[\nu_1, \nu_2, \nu_3, \nu_4]$. The fuzzy rules are based on the selected resolution level (first module) and the change in variance of the wavelet coefficients of the probe signal. The anytime auto-healing module published in [A. 16] ensures signal recovery from incomplete data. Further, this system has been extended with one more fuzzy module, performing similar manner as the other ones, that selects the appropriate wavelet function for the wavelet transform. Additionally, the reconstruction consists of searching for periodicity in the sequences stored in a buffer and performing the nearest neighbor interpolation method on the sparse samples [A. 16].

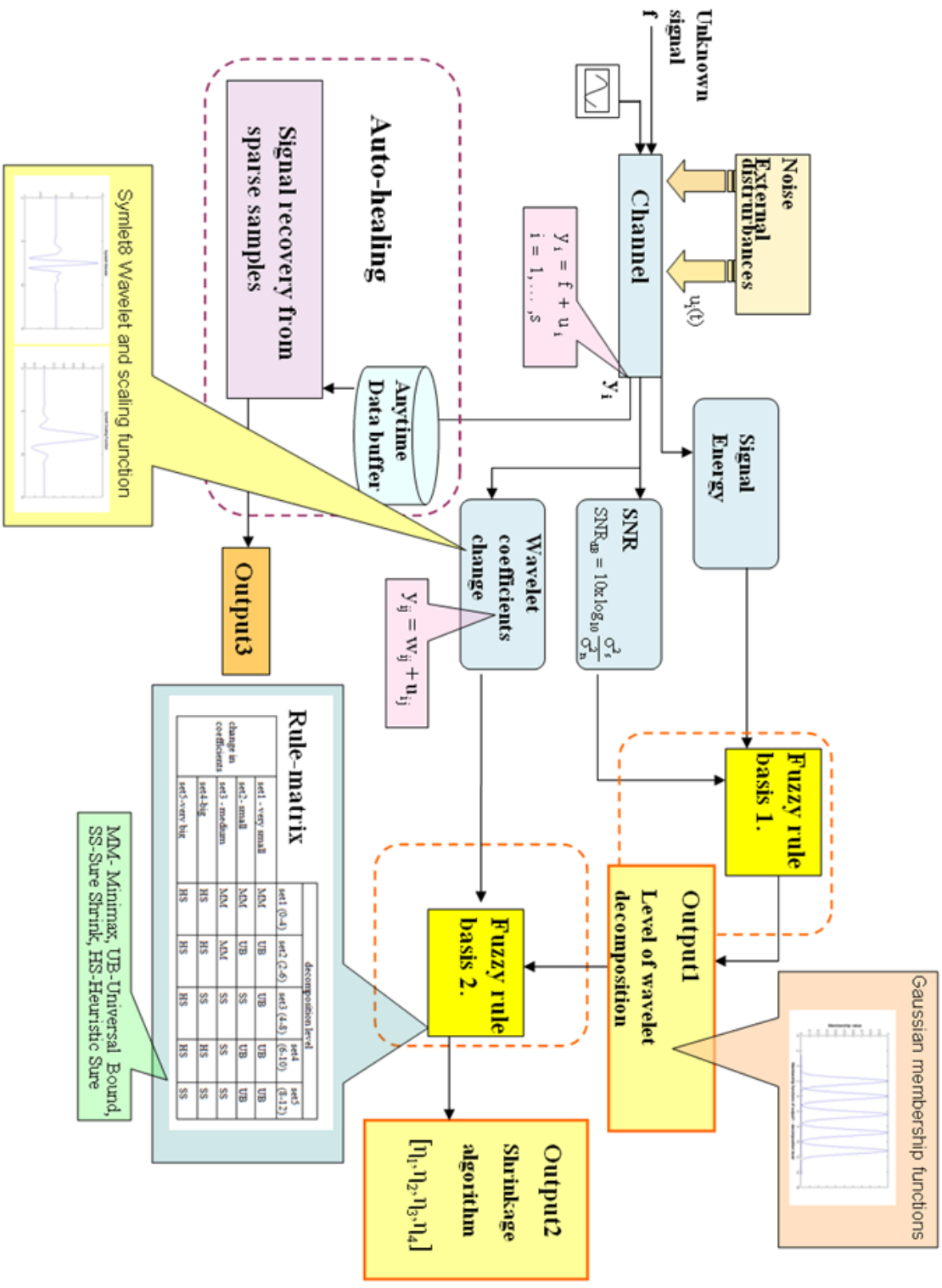

Figure 5.1: The scheme of the supervisory system

5.1.2 Improved Denoising

Regarding the key concepts in noise reducing applications another successful concept is minimizing the effects of outliers (extreme values or elements that deviate from the observation pattern [94]. Recent studies deal with the problem of outlier detection during the signal preprocessing. Most areas of engineering practice benefit from such algorithms, for instance monitoring and fault detection applications, data mining, etc. [95]. It is proven, that the robust local polynomial regression technique detects outliers excellently [96]. With this in mind, the present approach aims to utilize its advantageous feature for the thresholding operations in the wavelet domain.

The local polynomial regression (loess) procedure is also called locally weighted running line smoother. Extensions of the original method can be found in, e.g [96]. The principle of the method relies on approximation a function by fitting a regression surface to the data by determining a local neighbourhood of an arbitrary point t_0. These neighbouring points are weighted depending on their distance from the middlepoint. The closer points get larger w_i weights. The estimate is obtained by fitting a linear or quadratic polynomial using the weighted values from the neighbourhood. Detailed description of the procedure and the loess curve construction can be found in [97]. This method is based on least squares regression and it is known that this is vulnerable to outliers that can significantly degrade the result. For introducing robustness in the procedure an iterative reweighting according to the residuals is proposed with bisquare method [97]. Using the advantages of the latter procedure the proposed shrinkage approach includes the following steps. At first the signal is decomposed with the orthogonal wavelet functionsselected by the fuzzy supervisor. After the separation of the the detail and approximate coefficients of the signal a robust fitting is applied on the coefficients on each level. In this step the supervisor sets the appropriate parameters. Afterwards, the signal is reconstructed with inverse discrete wavelet transform. Thus, the realization of the new shrinkage procedure is the following [A. 15]: 1.) perform the discrete wavelet transform, 2.) fit the local polynomial regression curve on the coefficients with the w_i weights; 3.) get the residuals; 4.) get the median absolute value of the residuals; 5.) calculate the robust weights 6.) repeat step 1 ,7.) repeat step 3 to 6 until it converges; 8.) perform the IDWT.

5.1.3 Simulation Results

The performance of the proposed procedure has been tested on a one-dimensional signal corrupted with additive white Gaussian and impulse noises. The results have been compared with two other conventional shrinkage algorithms. The simulation is built by using Matlab8.

-	Improved Denoising	HeurSure	SureShrink	Minimax
SNR [dB] before	2.2294	2.2294	2.2294	2.2294
SNR [dB] after	23.5412	8.7601	3.6653	5.9276
RMSE	0.04159	0.5372	0.7621	0.3293
Level of Decomposition	1	4	8	4
Abs. max error	0.2289	0.8621	0.9535	31.1107
Elapsed time [s]	0.10196	0.0184	0.0127	0.0312

Table 5.1: The results of numerical simulations

The performance is measured by the root mean square error (RMSE) and the signal to noise ratio (SNR), calculated by the formula $SNR_{dB} = 10log\frac{\sigma_s^2}{\sigma_n^2}$, where σ_s^2 is the variation of the signal after denoising and σ_n^2 is the variation of the eliminated noise. The results are summarized in Table 5.1:

The performance of the robust fitting-based method can be seen in Fig. (5.2). The procedure precisely removes the noise and smooths the signal. Though, the HeurSure and the Minimax rule are faster (Table 5.1) and eliminate additive noise, but can not cope with impulse-type noise (Figs. 5.3). Since the reconstruction is not sufficient, further smoothing and outlier-eliminating processes are desired, which may increase the total elapsed time. With this in mind, the speed of the proposed procedure is acceptable.

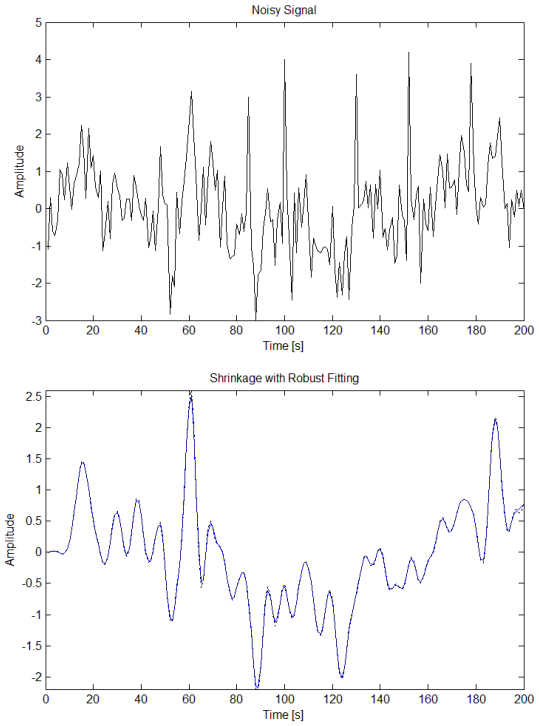

Figure 5.2: The original signal corrupted with noise (Upper Chart) and the result of denoising with the proposed method (Lower Chart, solid line - result of denoising, dotted line - original signal)

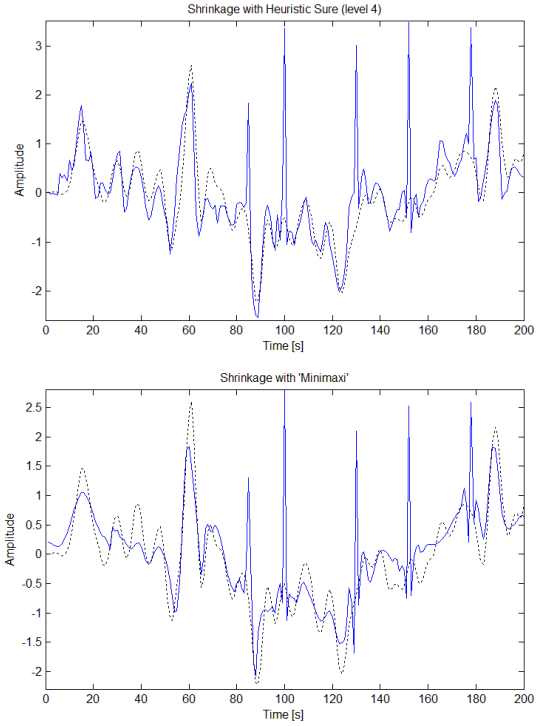

Figure 5.3: The performance of the HeurSure (Upper Chart) and the Minimax method (Lower Chart). (solid line - result of denoising, dotted line - original signal)

5.2 Thesis Statement IV.

I have introduced a new method that relies on the combination of anytime and other soft computing techniques for thresholding the coefficients in the wavelet transform domain. The proposed method combines the main advantages of multiresolution analysis and robust fitting. The anytime supervisory system supports the automatic wavelet shrinkage procedure. The wavelet function and the level of decomposition that are the most suitable in the given scenario and the parameters of the fitting are determined online by the fuzzy supervisory expert. The system applies orthogonal wavelet functions in order to significantly reduce the processing time of reconstruction.

Related publications: [A. 12], [A. 13], [A. 14], [A. 15], [A. 16]

Chapter 6

Adaptive Multi-round Smoothing based on the Savitzky - Golay Filter

The great majority of signal processing applications require advanced processing methods in order to achieve the desired precision of the result. In a particular class of tasks, for instance chemical spectroscopy, smoothing and differentiation is very significant. The details of the smoothing filters are well studied. In 1964 a great effort has been devoted to the paper of Savitzky and Golay, in which they introduced a particular type of low-pass filter, the so-called digital smoothing polynomial filter (DISPO) or Savitzky-Golay (SG) filter [98]. Its great advantage in contrast to the classical filters - that require the characterization and model of the noise process-, is that both the smoothed signal and the derivatives can be obtained by a simple calculation. Critical analysis and proposals for the modifications of the original method have been presented, for instance in [99][100]. The basis of their mehod is fitting a low degree polynomial in least squares sense on the samples within a sliding window. After, the new smoothed value of the centerpoint obtained from convolution. An ample number of papers dicussing its properties and possible improvements were written in [101][102][103][104][105][106][107][108]. The importance and applicability of a digital smoothing polynomial filter in chemometric algorithms are also well documented [109][110][111]. While, the frequency domain properties of SG-filters are revealed in [112][113][114][115]. Paper [116] concerns the properties of the SG digital differentiator filters and also the issue of the choice of filter length. In [117] the calculation of the filter coefficients for even-numbered data is addressed. Furthermore, the fractional-order SG differentiators have been investigated, as an illustration, by using the Riemann-Lioueville fractional order definition in the SG-filter. For example, the fractional order derivative can be calculated of corrupted signals as published in [118]. There are several sources and types of noise that may occur, for instance, eletronic noise, electromagnetic and electrostatic noise,

etc.[119]. However, it is commonly assumed that the noise is an additive white Gaussian noise (AWGN) process. In engineering practice often nonstationary, impulsive type disturbances, etc., can degrade the performance of the processing system. Since, for the noise removal issue of signals with a large spectral dynamics or with a high rate of change, the classical SG filtering is an unefficient method. Additionally, the performance depends on the appropriate selection of the polynomial order and the window length. The arbitrary selection of these parameters is difficult for the users. Usually the Savitzky-Golay filters perform well by using a low order polynomial with long window length or low degree with short window. This latter case needs the repetition of the smoothing. It has also been declared that the performance decreases by applying low order polynomial on higher frequencies. Nonetheless, it is possible to further improve the efficiency. With this goal, in this chapter I will describe a new adaptive smoothing approach based on the SG filtering technique that ensures acceptable performance independently of the type of noise process.

6.1 Brief Inroduction of the Mathematical Background behind the Savitzky-Golay Filter

In this section the premise behind the Savitzky–Golay filtering will be briefly outlined according to [120]. Firstly, let us consider equally spaced input data of n$\{x_j; y_j\}$, $j = 1, ..., n$. The smoothed values are derived from convolution, given by

$$g_i = \sum_{i=-m}^{m} c_i y_{k+i}, \tag{6.1}$$

where the window length $M = 2m + 1$, $i = -m, ..., \lambda, ..., m$, and λ denotes the index of the centerpoint. The k^{th} order polynomial P can be written as

$$P = a_0 + a_1(x - x_\lambda) + a_2(x - x_\lambda)^2 + ... + a_k(x - x_\lambda)^k \tag{6.2}$$

The goal is to calculate the coefficients of Eq. (6.1) by minimizing the fitting error in the least squares sense. The Jacobian matrix is as follows

$$J = \frac{\partial P}{\partial a} \tag{6.3}$$

The polynomial at $x = x_\lambda$ takes the value of a_0, so in order to evaluate the polynomial in the window we have to solve a system of M number of equations which can be written in matrix form

$$J \cdot a = y \tag{6.4}$$

Table 6.1: Some SG coefficients. $M = 2m + 1$ is the window length and k denotes the polynomial degree

Savitzky-Golay coefficients		
M	**k**	**Coefficients**
2*9	2	-0.0909 0.0606 0.1688 0.2338 0.2554 0.2338 0.1688 0.0606 -0.0909
	4	0.0350 -0.1282 0.0699 0.3147 0.4172 0.3147 0.0699 -0.1282 0.0350
2*11	3	-0.0839 0.0210 0.1026 0.1608 0.1958 0.2075 0.1958 0.1608 0.1026 0.0210 -0.0839
	5	0.0420 -0.1049 -0.0233 0.1399 0.2797 0.3333 0.2797 0.1399 -0.0233 -0.1049 0.0420

$$\begin{pmatrix} 1 & (x_{\lambda-m} - x_\lambda) & \cdots & (x_{\lambda-m} - x_\lambda)^k \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (x_{\lambda+m} - x_\lambda) & \cdots & (x_{\lambda+m} - x_\lambda)^k \end{pmatrix} \times \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} y_{\lambda-m} \\ \vdots \\ \vdots \\ \vdots \\ y_{\lambda+m} \end{pmatrix}$$

The coefficients are obtained from the normal equation as given below

$$J^T(Ja) = (J^T J)a \tag{6.5}$$

so

$$a = (J^T J)^{-1}(J^T y). \tag{6.6}$$

Since

$$P(x_\lambda) = a_0 = (J^T J)^{-1}(J^T y), \tag{6.7}$$

by replacing y with a unit vector in Eq.6.1 the c_0 coefficient can be calculated as

$$c_j = \sum_{i=1}^{k+1} |(J^T J)^{-1}|_{0i} J_{ij}. \tag{6.8}$$

With a size of $(2m + 1) \times (k + 1)$ the G matrix of the convolution coefficients

$$G = J(J^T J) = [g_0, g_1, ..., g_j]. \tag{6.9}$$

Fig. 6.1 displays the performance of the originall SG-filter. It can be seen that the smoothing is not precise. In order to address this problem, the following section will present an adaptive strategy.

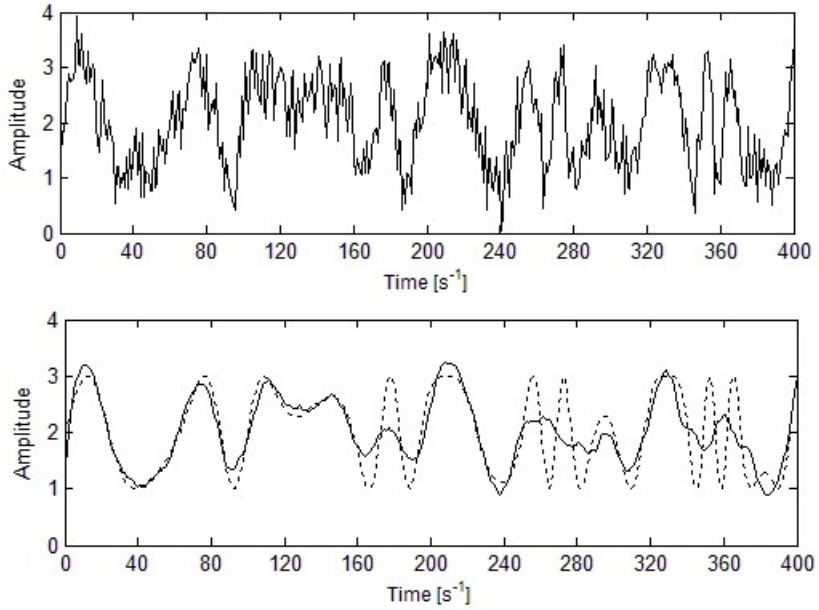

Figure 6.1: Performance of the original SG filter. Upper chart: signal with contaminating noise. Lower chart: dotted line - original signal, solid line - smoothed signal, $k = 3$, $M = 35$

6.2 Adaptive Multi-round Smoothing based-on the SG Filtering Technique

6.2.1 Multi-round Smoothing and Correction by the use of Fuzzy Rules

This new adaptive strategy aims setting automatically the suitable polynomial order and window length at the different frequency components of the signal [A. 18]. Hence, it is possible to avoid the undershoots and preserve the peaks that could be important from different data analysis aspects. Since we perform in the time domain, this method provides efficient results independently of the type of contaminating noise. At first, the classical Savitzky-Golay filtering is performed. Assuming that only the corrupted signal is available, this step serves for revealing the peaks, hence the window length and degree of the polynomial may be arbitrary. After the first smoothing, the coordinates of the local minimum and maximum points can be obtained. From now on, we can also define the distace vector d which contains the number of samples between two neighboring points of local minima and maxima. Then, the next step is the separation of the high- and low frequency components using the bordering points and setting the proper parameters for the smoothing. The window should match the scale of the signal and the polynomial degree should vary by depending on the framesize and frequency.

Since the next fuzzy relation can be defined between the section lengths [A. 17];

$$F(d_{max} >> \bar{d_R}) = \frac{1}{1 + e^{-(\delta_{max} - \bar{d_R})}} \in [0, 1] \tag{6.10}$$

where $\bar{\delta_R}$ stands for the average length of the sections in the current R parts of the signal, while $\delta_{max} = max(d)$ in the observed signal. If $g(d_{max}, \bar{d_R}) = 1$, the current part of the signal contains high freqency components. Hence, the following rules are applied [A. 18]:

if $1 > g(d_{max}, \bar{d_R}) > 0.9$ then $k = 5$, $M = nint(0.3\bar{\delta_R})$

if $0.89 > g(d_{max}, \bar{d_R}) > 0.75$ then $k = 4$, $M = nint(0.5\bar{\delta_R})$

if $0.75 > g(d_{max}, \bar{d_R}) > 0.45$ then $k = 3$, $M = nint(\bar{\delta_R})$

if $0.44 > g(d_{max}, \bar{d_R}) > 0.2$ then $k = 2$, $M = nint(0.5R_n)$

else $k = 1$, $M = nint(0.8R_n)$,

(6.11)

where R_n is the total number of samples of the R part, and we can assign the k and M values to each R part of the signal. In the rules the M values are rounded according to the nearest integer (nint). The values for the bounds have been determined according to the formula 2^k modified by experimental results.

6.2.2 New Parametric Weighting Function

The correction carried out with taking the linear approximation of the obtained signal. Then, it is extracted from the smoothed one. This step reveals the higher deviation, thus the next smoothing procedure can be modified according to its result. As we have the coordinates of the local minimum and maximum points and the vector d, we can easily fit a regression line on the points between two local extrema. In order to ensure the continuous joining of the lines we can perform this step by applying the Lagrange-multiplicator method given by

$$\sum_{x_i \in [x_1, x_2]} (m_1 x^{(i)} + b_1 - y^{(i)})^2 + \sum_{x_i \in [x_1, x_2]} (m_2 x^{(i)} + b_2 - y^{(i)})^2 \Rightarrow \min, \tag{6.12}$$

with the following constraints:

$$m_1 x_2 + b_1 - m_2 x_2 + b_2. \tag{6.13}$$

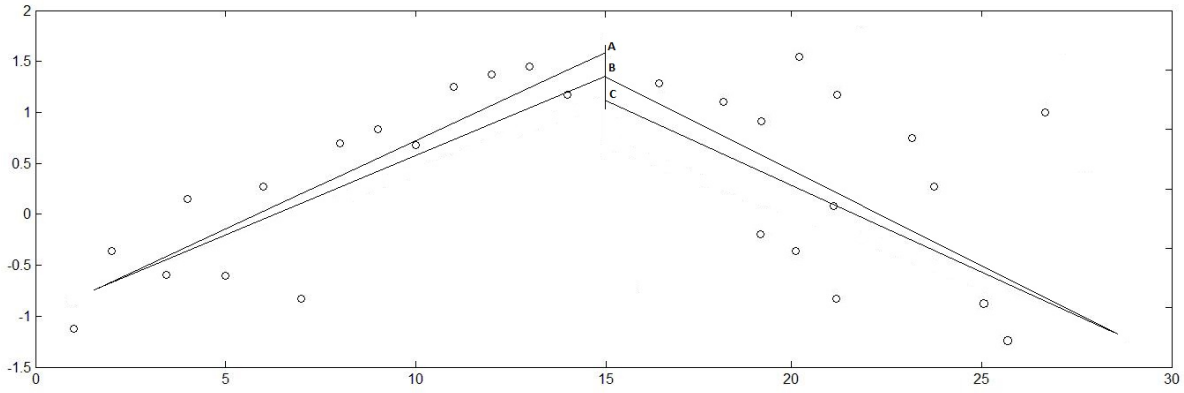

Figure 6.2: Illustration of problem of joining the regression lines

However, in some cases the peaks can contain the information of interest. Therefore, the form of the peak or valley should be processed with special care. To address this issue, a modified Shepard - method can be applied [A. 17]. Let us consider the points around the local extrema in radius r. The new values are calculated by weighting according to the neighboring points distance. There are several variations of the Shepard method [121], now let us consider the GMS ($Groundwater\ Modeling\ System$) form below

$$w_i = \frac{(\frac{d-d_i}{dd_i})^2}{\sum_{i=1}^{n} (\frac{d-d_i}{dd_i})^2}. \tag{6.14}$$

Eq. (6.14) can be trasformed into

$$w_i = \frac{(\frac{1-u_i}{u_i})^2}{\sum_{j=1}^{n} (\frac{u-u_j}{u_j})^2}, \tag{6.15}$$

in which $u_i(x) = \frac{d^i(x)}{d(x)}$. Now, using the similarity between the form of Eq. (6.15) and the $Dombi$ operator [122] we can define the following new parametric weighting function [A. 17]:

$$w_i = \frac{1}{1 + (\frac{u_i}{1-u_i})^2 \sum_{j=1} (\frac{1-u_j}{u_j})^{\lambda}} \tag{6.16}$$

in which the setting of λ and radius r (where it performs) have the effect on the smoothness of the result.

6.2.3 Simulation Results

The performance of the proposed method have been tested on a noisy signal (see, Fig. 6.3). The simulation is carried out by using Matlab8. Figure 6.4 shows the approximated signal after the first round. In Fig. 6.5 the resulted and the original signal can be seen after two rounds. It can be observed that the applied technique can efficiently recover the signal.

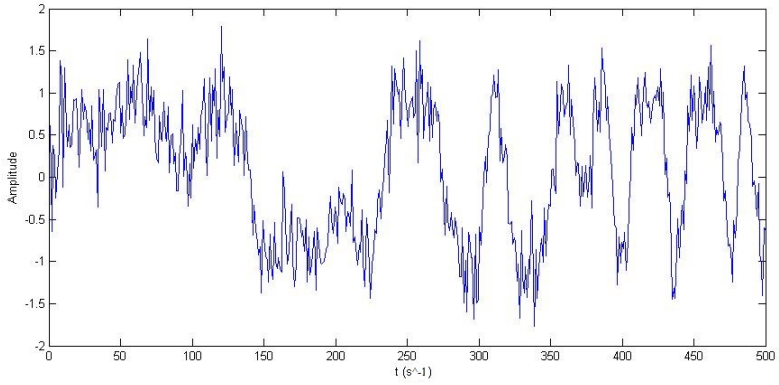

Figure 6.3: The noisy signal.

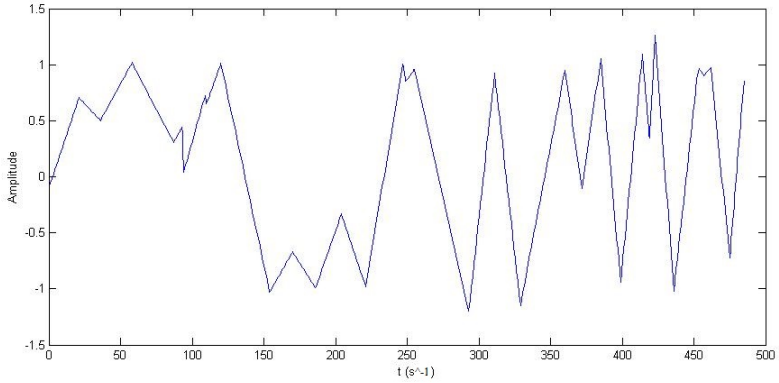

Figure 6.4: Approximation of the signal after the first round.

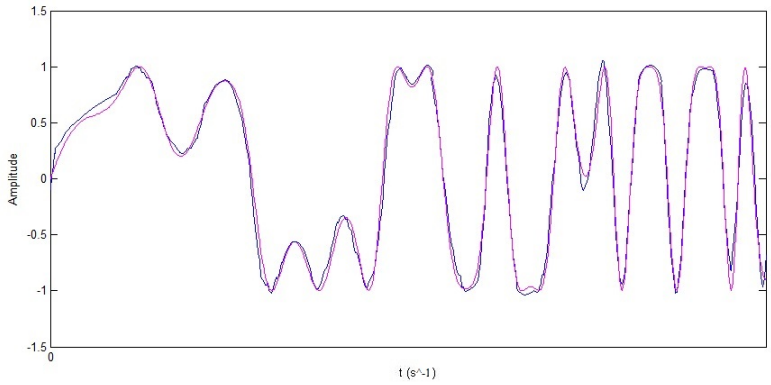

Figure 6.5: The recovered (blue) and the original (magenta) signal.

6.3 Thesis Statement V.

I have developed a new adaptive smoothing strategy based on the Savitzky-Golay filtering technique. The proposed method allows to evade the main difficulties of the original SG filter by automatically setting the smoothing parameters. Furthermore, for the precise reconstruction of the signal a multi round correction has been applied using the linear approximation of the signal. For the reconstruction of the peaks and valleys that may contain the important information, a new parametric weighting function has been introduced.

Related publications: [A. 17], [A. 18]

Chapter 7

Conclusions

Practical solutions of engineering problems involve model-integrated computing. Model-based approaches offer a very challenging way to integrate a priori knowledge into the procedure. Due to their flexibility, robustness and easy interpretability, the application of soft computing, in particular fuzzy and neural network-based models, may have an exceptional role at many fields. Especially in cases where the problem to be solved is highly nonlinear or when only partial, uncertain and/or inaccurate data is available. At the same, their usage can be so advantageous, it is still limited by their exponentially increasing computational complexity. Combining soft computing, non-conventional and novel data representation techniques is a possible way to overcome this difficulty.

The performance of a controller depends on the available form of the model, since my Thesis addresses novel data representation and control methods that are able to adaptively cope with usually imperfect, noisy or even missing information, the dynamically changing, possibly insufficient amount of resources and reaction (such as wavelet based multiresolution controllers, anytime control, Situational control, Robust Fixed Point Transformation (RFPT)-based control). The great majority of the adaptive nonlinear control design are based on Lyapunov's 2^{nd} or commonly referred to as the "Direct" method. The major defect of this method that it is mathematically complicated and it works with a large number of arbitrary adaptive control parameters and additionally the parameter identification process in certain cases is vulnerable if unknown external perturbations can disturb the system under control, etc. In the recent years the RFPT has been introduced for replacing the Lyapunov technique. Since, in this Thesis my first aim was dealing with the possibilities of the combination of the classical model-identification and the RFPT-based design in depth. I have proposed a new method that utilize the geometric interpretation provided by the Lyapunov-technique that can be directly used for parameter tuning. I have shown that these useful information can be obtained on the actual parameter estimation error by using the same feedback terms and equa-

tions of motion as the original methods. In order to improve the parameter tuning process, I have suggested the application of the Modified Gram-Schmidt Algoritm for the possible combination of the RFPT-based method with the *Modified Adaptive Inverse Dynamic Robot Controller (MAIDRC)* and the *Modified Adaptive Slotine-Li Robot Controller (MADSLRC)*. Besides, I have presented an even simpler tuning technique in the case of the Modified Adaptive Inverse Dynamics Robot Controller that also applies fixed point transformation-based tuning rule for parameter identification.

Afterwards, I have presented a systematic method for the generation of a new family of the Fixed Point Transformations, the so-called Sigmoid Generated Fixed Point Transformation (SGFPT) for the purposes of „Adaptive Dynamic Control" for nonlinear systems. At first, I have outlined the idea for the „Single Input - Single Output (SISO)" systems, then I have shown that it can be extended to „Multiple Input - Multiple Output (MIMO)" systems. Additionally, I have replaced the tuning method by a simple calculation in order to further simplify and improve the method.

I have proposed new advances regarding the „ *Sigmoid Generated Fixed Point Transformation (SGFPT)*". Also, I have described a new control strategy based on the combination of the "adaptive" and "optimal" control by applying time-sharing strategy in the SGFPT method, that supports error containment by cyclic control of the different variables. Further, I have introduced new improvements on SGFPT technique by introducing "Stretched Sigmoid Functions". The efficiency of the presented control solution have been confirmed by the adaptive control of an underactuated mechanical system. I have investigated the applicability of fuzzy approximation in the SGFPT-type control design and demonstrated the usability via simulation investigations. Furthermore, I have shown a new type of function for the SGFPT.

The other important issue that includes the maintenance of unwanted sensor noises that are mainly introduced by feedback into the system under control. Accordingly, in the development of a control system the signals of noisy measurements has to be addressed first so that more sophisticated signal pre-processing methods are required. Therefore, I have concerned the issue of well-adapted techniques for smoothing problems in the time domain and fitting data to parametric models. I have suggested new startegies for thresholding operations in the wavelet domain supported by anytime fuzzy supervisory system. I have investigated the Savitzky-Golay (SG) smoothing and differentiation filter. It has been proven that the performance of the classical SG-filter depends on the appropriate setting of the windowlength and the polynomial degree. The main limitations of the performance of this filter are the most conspicuous in processing of signals with high rate of change. In order to evade these deficiencies I have developed a new adaptive design to smooth signals based on the Savitzky-

Golay algorithm. The provided method ensures high precision noise removal by iterative multi-round smoothing. The signal approximated by linear regression lines and corrections are made in each step. Also, in each round the parameters are dynamically change due to the results of the previous smoothing. For supporting high precision reconstruction I have introduced a new parametric weighting function. Applicability of the Thesis have been confirmed by numerical simulations.

Chapter 8

Possible Targets of Future Research

Recently, $Non - conventional$ approaches has received much attention in the design of nonlinear adaptive control and signal processing. On account of the characteristics of Soft Computing techniques, such as flexibility and robustness, they have become fundamental tools in many areas. These methods are suitable for solving problems that are highly nonlinear or when only partial, uncertain data is available. In such situations, usual approaches are often impractical or computationally demanding. Since, my Thesis attempts to shed new light on Soft Computing, non-conventional and novel data representation techniques. In this Thesis I have presented new methods of adaptive control and signal processing that are able to adaptively cope with usually imperfect, noisy or even missing information. However, this research has thrown up many questions in need of further investigations. The presented findings suggest the following directions for future work; in Chapter 3 a systematic method has been presented for the generation of whole families of fixed point transformations, the so-called SGFPT. Considerable progress have been made with regard to the controller's performance by the use of the new types of functions. It is recommended that further investigations should target new methods for the automatic setting of these functions.

In Section 4.3 the enhancement of the SGFPT based control design by fuzzy approximation has been concerned. In the fixed point transformation instead of a unit matrix, a diagonal matrix with positive main diagonals was applied, that can be tuned to improve the convergence properties of the controller. It has been revealed, that its matrix elements can be tuned by observing little fluctuations in the convergence of the adaptive signal when these main diagonals are too big. Based on these observations, future research should focus on new tuning algorithms in order to further improve the convergence properties.

Regarding the results of combining the RFPT with neural networks and fuzzy modelling (see, [A. 19][A. 11]), a possible goal can be the investigations of the combination of RFPT method and wavelet technique in the control of strongly nonlinear systems. Early foundings have

been published in [A. 21] about the applicability of the Fixed Point Transformation-based Adaptive Control design for automatic control of the depth of hypnosis during surgical operation. The here presented technique regulates the WAV_{CNS} index as the only measurable variable by controling the intravenous propofol administration. Therefore a possible goal of future work should aim revealing the links between the SGFPT-type control and Wavelet Theory focusing on the novel methods for dynamical problems.

List of Figures

2.1 The trajectory tracking error of the AIDC in the case free of external disturbances (upper chart) and in the case of disturbance forces (3^{rd} order spline functions of time) (lower chart) 20
2.2 The disturbance forces pertaining to the lower chart of Fig. 2.1 20
2.3 Tuning of parameter $\Theta_4 \equiv \hat{k}$ without (upper chart) and with (lower chart) external disturbances 21
2.4 Tuning the other parameters in Θ without (upper chart) and with (lower chart) external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines] 21
2.5 The trajectory tracking error of the AIDC with modified tuning in the case free of external disturbances 22
2.6 Tuning of parameter $\Theta_4 \equiv \hat{k}$ of the AIDC with modified tuning without external disturbances 22
2.7 Tuning the other parameters in Θ of the AIDC with modified tuning without external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines] 22
2.8 The trajectory tracking error of the AIDC (upper chart) with modified tuning and limited external disturbances (lower chart) 23
2.9 Tuning the parameters in Θ of the AIDC with modified tuning with reduced external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_4 \equiv \hat{k}$: red, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines] 24
2.10 The trajectory tracking error of the RFPT-supported AIDC (upper chart) with modified tuning and considerable external disturbances (lower chart) 26
2.11 Tuning the parameters in Θ of the RFPT-supported AIDC with modified tuning with considerable external disturbances [$\Theta_1 \equiv \hat{m}$: black, $\Theta_2 \equiv \hat{\mu}$: blue, $\Theta_3 \equiv \hat{\mu}\hat{c}$: green, $\Theta_4 \equiv \hat{k}$: red, $\Theta_5 \equiv \hat{\beta}$: magenta, and $\Theta_6 \equiv \hat{\lambda}$: ocher lines] 26

2.12 The phase trajectory tracking of the RFPT-supported AIDC with modified tuning and considerable external disturbances 27
2.13 The tracking error in the lack of unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines] 32
2.14 Tuning of the adaptive parameters in the lack of unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[Θ_1: solid, Θ_2: dashed, Θ_3: dense dash, Θ_4: dash-dot, and Θ_5: dash-dot-dot lines] 33
2.15 The tracking error under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines] 33
2.16 Tuning of the adaptive parameters under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[Θ_1: solid, Θ_2: dashed, Θ_3: dense dash, Θ_4: dash-dot, and Θ_5: dash-dot-dot lines] 34
2.17 The phase trajectories under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines] 35
2.18 The trajectory tracking under unknown disturbances: with modified tuning *without RFPT* (upper chart), and modified tuning *with RFPT* (lower chart)[q_1: solid, q_2: dashed, q_3: dense dash lines] 35
2.19 The second time-derivatives of generalized coordinate q_3 with modified tuning and RFPT-based adaptation (zoomed excerpt in the lower chart) [$\ddot{q}_3$ (realized): solid, $\ddot{q}_3^{Des}$ ("desired"): dashed, $\ddot{q}_3^{Req}$ (adaptively deformed): dense dash lines] 36
2.20 The Trajectory Tracking (iteration using initial value 0) 39
2.21 The Trajectory Tracking (using result of previous cycle as initial value) . . 40
2.22 The Trajectory Tracking Error (iteration using initial value 0) 40
2.23 The Trajectory Tracking Error (using result of previous cycle as initial value) 41
2.24 The Phase Trajectory Tracking (iteration using initial value 0) 41
2.25 The Phase Trajectory Tracking (using result of previous cycle as initial value) 42
2.26 The Error of the Known Term (iteration using initial value 0) 42
2.27 The Error of the Known Term (using result of previous cycle as initial value) 43
2.28 The Dynamic parameters (iteration using initial value 0) $\hat{\theta}_1 - black, \hat{\theta}_2 - blue, \hat{\theta}_3 - green, \hat{\theta}_4 - red, \hat{\theta}_5 - orange$ 43

2.29 The Dynamic parameters (using result of previous cycle as initial value) $\hat{\theta}_1 - black, \hat{\theta}_2 - blue, \hat{\theta}_3 - green, \hat{\theta}_4 - red, \hat{\theta}_5 - orange$ 44
2.30 The Dynamic Parameters (iteration using initial value 0) $\hat{\theta}_6 - black, \hat{\theta}_7 - blue, \hat{\theta}_8 - green, \hat{\theta}_9 - red, \hat{\theta}_{10} - orange$ 44
2.31 The Dynamic parameters (using result of previous cycle as initial value) $\hat{\theta}_6 - black, \hat{\theta}_7 - blue, \hat{\theta}_8 - green, \hat{\theta}_9 - red, \hat{\theta}_{10} - orange$ 45
2.32 The 2^{nd} Time Derivatives (iteration using initial value 0) 45
2.33 The 2^{nd} Time Derivatives (using result of previous cycle as initial value) . . 46

3.1 The basic idea of fixed point generation by the use of a sigmoid 49
3.2 The fixed points of $F(x) \stackrel{def}{=} g^{-1}(g(x) - K) + D$ 49
3.3 Schematic description of the iteration $x_{n+1} \stackrel{def}{=} F(x_n)$ as it converges to the attractive fixed point [$g(x_n) - K = g(x_{n+1} - D)$ corresponds to $x_{n+1} = g^{-1}(g(x) - K) + D$] 49
3.4 Example for slow monotonic convergence to the solution of the control task: $A = -0.5$ 50
3.5 Example for fast monotonic convergence to the solution of the control task: $A = -2$ 51
3.6 Example for precursor oscillations (non-monotonic convergence to the solution of the control task): $A = -3.12$ 51
3.7 Example for chaotic, divergent oscillations: $A = -6$ 51
3.8 Trajectory tracking for v in the non-adaptive (LHS) and the adaptive (RHS) cases: v^N: black solid, w^N: blue dashed, v: green dash dot, w: red dotted lines (LHS) 53
3.9 Trajectory tracking error for v in the non-adaptive (LHS) and the adaptive (RHS) cases: $v^N - v$: black solid, (in the adaptive case the error of w has been removed from the chart) 53
3.10 The phase trajectories for variable v the non-adaptive (LHS) and the adaptive (RHS) cases: $\dot{v}$ vs. v: nominal: solid blue, realized: red dotted lines 54
3.11 The phase trajectories for variable w the non-adaptive (LHS) and the adaptive (RHS) cases when only the trajectory of v is under control: $\dot{w}$ vs. w: nominal: solid blue, realized: red dotted lines 54
3.12 The "Desired" (black solid), the "Deformed" (blue dotted) (without deformation it exactly is identical to the "Desired" line), and the "Realized" (green dashed) time-derivatives for variable v in the non-adaptive (LHS), and the adaptive (RHS) cases 54
3.13 The control signal in the non-adaptive (LHS) and the adaptive (RHS) cases 55

3.14 Trajectory tracking for v in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: v^N: black solid, w^N: blue dashed, v: green dash dot, w: red dotted lines (LHS) 55
3.15 Trajectory tracking error for v in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $v^N - v$: black solid, $w^N - w$: blue dotted lines 55
3.16 The phase trajectories for variable v the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $\dot{v}$ vs. v: nominal: solid blue, realized: red dotted lines 56
3.17 The phase trajectories for variable w the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances: $\dot{w}$ vs. w: nominal: solid blue, realized: red dotted lines 56
3.18 The "Desired" (black solid), the "Deformed" (blue dotted) (without deformation it exactly is identical to the "Desired" line), and the "Realized" (green dashed) time-derivatives for variable v in the non-adaptive (LHS), and the adaptive (RHS) cases, under external disturbances. 56
3.19 The control signal in the non-adaptive (LHS) and the adaptive (RHS) cases, under external disturbances. 57
3.20 The convergence to the desired values for $A = 0.20$ (at the LHS) and $A = 0.27425$ (at the RHS): f_1^{Des}: magenta, f_2^{Des}: ocher, x_1: black, x_2: blue, f_1: green, f_2: red lines 61
3.21 Trajectory tracking in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green, q_2^{Nom} $[rad]$: red, q_3^{Nom} $[m]$: ocher lines] 62
3.22 Trajectory tracking error in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green lines] 63
3.23 The phase trajectories for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [for q_2: black, q_3: blue, q_2^{Nom}: red, q_3^{Nom}: ocher lines] . . . 63
3.24 The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines] . . 64
3.25 The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines] 64
3.26 The trajectory of the "driving arm" q_1 for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases 64

3.27 The *"desired"*, *"adaptively deformed"*, and the *"realized"* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) for $A = -3.125$ [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines] . . 65
3.28 Trajectory tracking in the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green, q_2^{Nom} $[rad]$: red, q_3^{Nom} $[m]$: ocher lines] 67
3.29 Trajectory tracking error in the non-adaptive (at the LHS) and the adaptive (at theRHS) cases [q_2 $[rad]$: black, q_3 $[m]$: green lines] 68
3.30 The phase trajectories for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [for q_2: black, q_3: blue, q_2^{Nom}: red, q_3^{Nom}: ocher lines] . . . 68
3.31 The *desired*, *adaptively deformed*, and the *realized* 2^{nd} time-derivatives for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines] . . 68
3.32 The *desired*, *adaptively deformed*, and the *realized* 2^{nd} time-derivatives for the adaptive case (zoomed in excerpts) [$\ddot{q}_2^{Des}$ $[rad]$: black, $\ddot{q}_3^{Des}$ $[m]$: green, $\ddot{q}_2^{Def}$: red, $\ddot{q}_3^{Def}$: ocher, $\ddot{q}_2$: brown, $\ddot{q}_3$: blue lines] 69
3.33 The trajectory of the driving arm q_1 for the non-adaptive (at the LHS) and the adaptive (at the RHS) cases 69
3.34 The calculated parameter A vs. time 69

4.1 Example of the *"swinging paradigm"*: in the *underactuated system* axle q_1 has no driving torque, i.e. the appropriate generalized force $Q_1\,[N] \equiv 0$. The driving torque of axle q_2 i.e. $Q_2\,[N]$ is used for the realization of a compromise in approximately tracking a nominal trajectory $q_1^N(t) \neq 0$ and $q_2^N(t) \equiv 0$. (This latter restriction is introduced for saving the body of the swinging child.) 74
4.2 The structure of the controller and the simulation 74
4.3 Non-adaptive trajectory tracking; top: $q_1^N(t)$: black line, $q_1(t)$: green line; bottom: $q_2^N(t)$: green line, $q_2(t)$: red line 75
4.4 Adaptive trajectory tracking; top: $q_1^N(t)$: black line, $q_1(t)$: green line; bottom: $q_2^N(t)$: green line, $q_2(t)$: red line 75
4.5 Time-dependence of $\ddot{q}_1$ (zoomed-in excerpts): non-adaptive control: top, adaptive control bottom (color codes: black line: $\ddot{q}_1^N(t)$ nominal, blue line: $\ddot{q}_1^{Des}$ kinematically prescribed "desired", green line: $\ddot{q}_1^{Def}$ adaptively deformed, red line: $\ddot{q}_1$ realized (simulated), yellow line: the timer: for 1 q_1 is under control, for 2 q_2 is under control) 76

4.6 Time-dependence of $\ddot{q}_2$ (zoomed-in excerpts): non-adaptive control: top, adaptive control bottom (color codes: black line: $\ddot{q}_2^N(t)$ nominal, blue line: $\ddot{q}_2^{Des}$ kinematically prescribed "desired", green line: $\ddot{q}_2^{Def}$ adaptively deformed, red line: $\ddot{q}_2$ realized (simulated), yellow line: the timer: for 1 q_1 is under control, for 2 q_2 is under control) 76
4.7 The scheme of the upside-down pendulum system 77
4.8 The structure of the controller and the simulation 79
4.9 Trajectory tracking and tracking error in the the adaptive case 79
4.10 Trajectory tracking and tracking error in the the adaptive cases -zoomed . . 80
4.11 Trajectory tracking and tracking error in the the non-adaptive cases -zoomed 80
4.12 Time-dependence of $\ddot{q}_2$ in case of adaptive control (color codes: black line:$\ddot{q}_2^{Des}$ kinematically prescribed "desired", blue line: $\ddot{q}_2^{Def}$ adaptively deformed, green line: $\ddot{q}_2$ realized (simulated) 80
4.13 Time-dependence of $\ddot{q}_2$ in case of adaptive control -zoomed (color codes: black line:$\ddot{q}_2^{Des}$ kinematically prescribed "desired", blue line: $\ddot{q}_2^{Def}$ adaptively deformed, green line: $\ddot{q}_2$ realized (simulated) 81
4.14 Q1 [N] vs. time in the adaptive case 81
4.15 Q1 [N] vs. time in the non-adaptive case 81
4.16 Trajectory tracking and tracking error of the non-adaptive controller for the *"affine model"* 84
4.17 Trajectory tracking and tracking error of the adaptive controller for the *"affine model"* 85
4.18 The $\ddot{q}$ values of the adaptive controller for the *"affine model"* 85
4.19 The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"affine model"* 85
4.20 Functions $\sin(x)$ (black), $\mu_s(x)$ (blue), $\cos(x)$ (green), and $\mu_c(x)$ (red) . . . 86
4.21 Trajectory tracking and tracking error of the non-adaptive controller for the *"soft computing-based model"* 87
4.22 Trajectory tracking and tracking error of the adaptive controller for the *"soft computing-based model"* 87
4.23 The $\ddot{q}$ values of the adaptive controller for the *"soft computing-based model"* 87
4.24 The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"soft computing-based model"* 88
4.25 Trajectory tracking and tracking errorof the non-adaptive controller for the *"soft computing-based model"* 89

4.26 Trajectory tracking and tracking error of the adaptive controller for the *"fully soft computing-based model"* 89
4.27 The $\ddot{q}$ values of the adaptive controller for the *"fully soft computing-based model"* 89
4.28 The tuned parameters of the adaptive controller and the content of the forgetting buffer for the *"fully soft computing-based model"* 90

5.1 The scheme of the supervisory system 95
5.2 The original signal corrupted with noise (Upper Chart) and the result of denoising with the proposed method (Lower Chart, solid line - result of denoising, dotted line - original signal) 98
5.3 The performance of the HeurSure (Upper Chart) and the Minimax method (Lower Chart). (solid line - result of denoising, dotted line - original signal) 99

6.1 Performance of the original SG filter. Upper chart: signal with contaminating noise. Lower chart: dotted line - original signal, solid line - smoothed signal, $k = 3$, $M = 35$ 104
6.2 Illustration of problem of joining the regression lines 106
6.3 The noisy signal. 107
6.4 Approximation of the signal after the first round. 107
6.5 The recovered (blue) and the original (magenta) signal. 108

Chapter 9

References

Publications Strictly Related to the Thesis

[A. 1] Dineva, A., Várkonyi-Kóczy, A. Tar, J.K.: *"Combination of RFPT-based Adaptive Control and Classical Model Identification"*, In Proc. of the IEEE 12^{th} Int. Symp. on Applied Machine Intelligence and Informatics (SAMI 2014), 2014, Herlány, Slovakia, 2014, pp. 35-40, **2014**

[A. 2] Tar, J.K., Rudas, I., Dineva, A. and Várkonyi-Kóczy A.: *"Stabilization of a Modified Slotine-Li Adaptive Robot Controller by Robust Fixed Point Transformations"*, In Proc. of Recent Advances in Intelligent Control, Modelling and Simulation, 2014, Cambridge, MA, USA, 2014, pp. 35-40, **2014**

[A. 3] Tar, J.K., Bitó, J., Várkonyi-Kóczy, A. and Dineva, A.: *"Symbiosys of RFPT-Based Adaptivity and the Modified Adaptive inverse Dynamics Controller"*, In: Advances in Soft Computing, Intelligent Robotics and Control (Eds. J. Fodor and R. Fullér, Springer Heidelberg, London, New York, 2014, pp. 95-106, **2014**

[A. 4] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Application of Fixed Point Transformation to Classical Model Identification using New Tuning Rule"*, Accepted for publication for the 15^{th} IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI), to be held in Herlány, Slovakia, between 26-28 January, **2017**

[A. 5] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A.: *"Novel Generation of Fixed Point Transformation for the Adaptive Control of a Nonlinear Neuron Model"*, In Proc. of the IEEE International Conference on System, Man, and Cybernetics (SMC2015), 2015, 9-12 October, HongKong, China, pp. 987-992, **2015**

[A. 6] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO Systems"*, In Proc. of the IEEE

19^{th} International Conference on Intelligent Engineering Systems (INES2015), September 3-5, 2015, Bratislava, Slovakia, pp. 135-140, **2015**

[A. 7] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Replacement of Parameter Tuning with Simple Calculation in Adaptive Control Using "Sigmoid Generated Fixed Point Transformation"*, In Proc. of the IEEE International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2015.09.17-2015.09.19., pp. 173-178, **2015**

[A. 8] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K. and Piuri, V.: *"Intelligent Neural Network Design for Nonlinear Control using Simultaneous Perturbation Stochastic Approximation (SPSA) Optimization"*, **JAPANESE JOURNAL OF APPLIED PHYSICS** (JJAP) CP, **4**, 011612-1 - 011612-6, **2016**

[A. 9] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Adaptive Control of Underactuated Mechanical Systems using Improved Sigmoid Generated Fixed Point Transformation and Scheduling Strategy"*, In Proc. of the IEEE 14^{th} International Symposium on Applied Machine Intelligence and Informatics, (SAMI 2016), pp. 193-197, **2016**

[A. 10] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Sigmoid Generated Fixed Point Transformation Control Scheme for Stabilization of Kapitza's Pendulum System"*,In Proc. of the IEEE 20^{th} Jubilee International Conference on Intelligent Engineering Systems: INES 2016, 2016.06.30-07.02., Budapest, Hungary, pp. 213-218, **2016**

[A. 11] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Adaptive Controller using Fuzzy Modeling and Sigmoid Generated Fixed Point Transformation"*, In Proc. of the IEEE 8^{th} International Conference on Intelligent Systems, (IS'16), Sofia, Bulgaria, 04.09.-07.09, 2016, pp. 522-527, **2016**

[A. 12] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: *"Wavelet-based Technique for Feedback Control of Uncertain Systems – A Case Study"*, In Proc. of the IEEE 14^{th} International Symposium on Computational Intelligence and Informatics,(CINTI 2013), Budapest, Hungary, 2013.11.19-2013.11.21, 2013, pp. 281-285, **2013**

[A. 13] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: *"Compactly Supported Wavelet Based Multiresolution PID Controller"*, In Proc. of the 12^{th} Int. Conf. on Global Research and Education in Intelligent Systems, Inter-academia'2013, Sofia, Bulgaria, 2013.09.23-2013.09.27., pp. 281-285, **2013**

[A. 14] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: *"Fuzzy Expert System for Automatic Wavelet Shrinkage Procedure Selection for Noise Suppression"*, In Proc. of the 18^{th} International Conference on Intelligent Engineering Systems (INES 2014), 2014.07.03-2014.07.05., Tihany, Hungary, pp.163-168, **2014**

[A. 15] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: *"Improved Denoising with Robust*

Fitting in the Wavelet Transform Domain", L. M. Camarinha-Matos and T. A. Baldissera and G. Di-Orio G and F. Marques (Eds.), IFIP Advances in Information and Communication Technology; Technological Innovation for Cloud-Based Engineering System **450**, Cham: Springer, pp. 179-187, **2015**

[A. 16] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: *"Anytime Fuzzy Supervisory System for Signal Auto-Healing"* **ADVANCED MATERIALS RESEARCH, 1117**, Trans Tech Publications, Switzerland, doi:10.4028/www.scientific.net/AMR.1117.269, pp. 269-272, **2015**

[A. 17] Dombi, J., Dineva, A.: *"Adaptive Multi-round Smoothing based on the Savitzky-Golay Filter "*, accepted for publication in: Advances in Intelligent Systems and Computing, Springer (ISSN 2194-5357), **2017**

[A. 18] Dombi, J., Dineva, A.: *"Adaptive Savitzky-Golay Filtering and Its Applications "*, In Press: **INTERNATIONAL JOURNAL OF ADVANCED INTELLIGENCE PARADIGMS (IJAIP)**, ISSN: 1755-0386, Inderscience Publishers, Switzerland, **2017**

Further Publications of the Author

[A. 19] Dineva, A., Várkonyi-Kóczy, A., Piuri, V. and Tar, J.K.: *"Performance Enhancement of Fuzzy Logic Controller using Robust Fixed Point Transformation"*, In: R. Jablonski, R. Szewczky (Eds.), Advances in Intelligent Systems and Computing 519; Recent Global Research and Education: Technological Challanges. Berlin, Heidelberg: Springer Verlag, 2017, pp. 411-418, **2017**

[A. 20] Dineva, A., Várkonyi-Kóczy, A., Piuri, V. and Tar, J.K.: *"Point Cloud Processing with the Combination of Fuzzy Information Measure and Wavelets"*, accepted for publication in: Advances in Intelligent Systems and Computing, Springer **2016**

[A. 21] Dineva, A., Tar, J.K., Várkonyi-Kóczy, A. and Piuri, V.: *"Adaptive Controller Using Fixed Point Transformation for Regulating Propofol Administration Through Wavelet-based Anesthetic Value"* In Proc. of the IEEE International Symposium on Medical Measurements and Applications, (MeMeA 2016) , 05.15-05.18, Benevento, Italy, pp. 650-655, **2016**

[A. 22] Dineva, A., Nagy, I.: *"Adaptiv Wavelet-alapú Zajcsökkentö Eljárás Mobilrobot Távérzékelési Adatfeldolgozó Rendszerében"*, MÜSZAKI TUDOMÁNYOS KÖZLEMÉNYEK **5:(5)**, pp. 129-132, **2016**

[A. 23] Dineva, A., Várkonyi-Kóczy, A., Tar, J.K.: "Wavelet-based Noise Removal Technique for Remotely Sensed Data", Repüléstudományi Közlemények, **2015:(3)**, pp. 149-158, **2015**

[A. 24] Kósi, K., Dineva, A., Tar, J.K.: *"Increased Cycle Time Achieved by Fractional*

Derivatives in the Adaptive Control of the Brusselator Model" In Proc. of the IEEE 11^{th} International Symposium on Applied Machine Intelligence and Informatics (SAMI 2013), 2013.01.31-2013.02.02., Herlány, Slovakia, pp. 65-70, **2013**

[A. 25] Nagy, I., Várkonyi-Kóczy, A. and Dineva, A.: *"GA Optimization and Parameter Tuning at the Mobile Robot Map Building Process "* In Proc of the IEEE 9^{th} International Conference on Computational Cybernetics (ICCC 2013), 2013.07.08-2013.07.10., Tihany, Hungary, pp. 333-338, **2013**

[A. 26] Várkonyi-Kóczy, A., Tusor, B., Dineva, A.: *"Determination of the Complexity Fitted Model Structure of Radial Basis Function Neural Networks"* In Proc. of the IEEE 17^{th} International Conference on Intelligent Engineering Systems (INES2013), 2013.06.19-2013.06.21, San Jose, Costa Rica, pp. 237-242, **2013**

Bibliography

[1] M. Kratmüller. Combining fuzzy/wavelet adaptive error tracking control design. *Acta Polytechnica Hungarica*, 7(4):115–137, 2010.

[2] A.R. Várkonyi-Kóczy. *Model Based Anytime Soft Computing Approaches in Engineering Applications, Balas, V., J. Fodor, A.R. Várkonyi-Kóczy (eds.), Soft Computing Based Modeling in Intelligent Systems (Ser. Studies in Computational Intelligence), pp. [4] 63-92*. Springer Verlag, Berlin, Heidelberg, 2009.

[3] A. R. Várkonyi-Kóczy. Using ts fuzzy models in anytime control systems. In *In.: Valentina E Balas (Ed.), SOFA 2010 - 4th International Workshop on Soft Computing Applications, Proceedings. Arad, Romania. 2010.07.15-2010.07.17. Seattle: IEEE, (ISBN:9781424479849)*, pages 215–220, 2010.

[4] A. R. Várkonyi-Kóczy. State dependant anytime control methodology for non-linear systems. *JOURNAL OF ADVANCED COMPUTATIONAL INTELLIGENCE AND INTELLIGENT INFORMATICS 12:(2)*, pages 198–205., 2008.

[5] Annamária R. Várkonyi-Kóczy. *Methods of Computational Intelligence for Modeling and Data Representation of Complex Systems (DSc Dissertation)*. Hungarian Academy of Sciences, Budapest, Hungary, 2008.

[6] L. Madarász, R. Andoga, L. Főző, and T. Lázár. *Situational control, modeling and diagnostics of large scale systems, In I. Rudas, J. Fodor, J. Kacprzyk (eds.) Towards Intelligent Engineering and Information Technology pp. 153-164*. Springer Verlag, Heidelberg, 2009.

[7] A. R. Várkonyi-Kóczy and I.J. Rudas. Ts fuzzy modeling based anytime control methodology for situational control. In *2011 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Hangzhou, China, 2011.05.10-2011.05.12, Piscataway: IEEE Computer Society, ISBN:978-1-4244-7935-1*, pages 1777–1782, 2011.

[8] J.K. Tar. *Adaptive Control of Smooth Nonlinear Systems Based on Lucid Geometric Interpretation (DSc Dissertation)*. Hungarian Academy of Sciences, Budapest, Hungary, 2012.

[9] António E. Ruano and A. R. Várkonyi-Kóczy, editors. *New Advances in Intelligent Signal Processing, (Studies in Computational Intelligence; 372.)*. Berlin; Heidelberg: Springer Verlag, (ISBN:978-3-642-11738-1), 2011.

[10] A. R. Várkonyi-Kóczy. Observer based iterative fuzzy and neural network model inversion for measurement and control applications. *In: Rudas I., Fodor J., Kacprzyk J. (Eds.), Towards Intelligent Engineering and Information Technology. (Studies in Computational Intelligence; 243.) ISBN:9783642037368, Berlin; Heidelberg: Springer-Verlag*, pages 681–702, 2009.

[11] Vincenzo Piuri. Design of fault-tolerant distributed control systems. *IEEE Transactions on Instrumentation and Measurement (ISSN: 0018-9456)*, pages 257–264, 1994.

[12] V. Piuri, C. Alippi, and C. de Russis. A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, ISSN: 1094-6977*, pages 259–268, 2003.

[13] Dhiraj Ahuja and Sunil Kumar. Adaptive fuzzy wavelet network control design for nonlinear systems. *International Journal of Advanced Technology & Engineering research (IJATER)*, 3(1):148–157, 2013.

[14] A. Soumelidis, F. Schipp, and J. Bokor. On hyperbolic wavelets. *Preprints of the 18th IFAC World Congress, S. Bittandi, A. Cenedese, and S. Zampieri, Eds., Milano, Italy, August 28 - Sep. 2 2011*, pages 2309–2314, 2011.

[15] C.P. Bernard and J.-J.E. Slotine. Adaptive control with multiresolution bases. *Proceedings of the 36th IEEE Conference on Decision and Control, 10-12 Dec 1997, San Diego, CA*, 4:3884–3889, 1997.

[16] A. Lyapunov. *A general task about the stability of motion. (in Russian)*. Ph.D. Thesis, University of Kazan, Tatarstan (Russia), 1892.

[17] J.K. Tar, J.F. Bitó, and I.J. Rudas. Replacement of Lyapunov's Direct Method in Model Reference Adaptive Control with Robust Fixed Point Transformations. *In Proc. of the* 14^{th} *IEEE Intl. Conf. on Intelligent Engineering Systems, Las Palmas of Gran Canaria, Spain*, pages 231–235, 2010.

[18] J.K. Tar, J.F.Bitó, I.J. Rudas, and Á. Szeghegyi. Formal application of the Symplectic Group for Soft Computing-based control of mechanical devices free of direct physical interpretation. *In Proc. of the IEEE Intl. Conf. on Intelligent Engineering Systems (INES), Helsinki, Finland*, pages 335–340, 2001.

[19] J.K. Tar, I.J. Rudas, J.F. Bitó, and K. Jezernik. A generalized Lorentz group-based adaptive control for DC drives driving mechanical components. *In Proc. of the* 10^{th} *Intl. Conf. on Advanced Robotics (ICAR) Budapest*, pages 299–305, 2001.

[20] J.K. Tar, I.J. Rudas, K. Kozłowski, and T. Ilkei. Application of the Symplectic Group in a novel branch of Soft Computing for controlling of electro-mechanical devices. *In Proc. of the* 2^{nd} *Intl. Workshop On Robot Motion And Control, RoMoCo, Bukowy Dworek, Poland*, pages 71–77, 2001.

[21] J.K. Tar, I.J. Rudas, J.F. Bitó, and K. Kozłowski. A new approach in computational cybernetics based on the Modified Renormalization Algorithm guaranteeing complete stability in the control of a wide class of physical systems. *In Proc. of the* 6^{th} *IEEE Intl. Conf. on Intelligent Engineering Systems (INES), Opatija, Croatia*, pages 19–24, 2002.

[22] I. Sekaj and V. Veselý. Robust output feedback controller design: Genetic Algorithm approach. *IMA J Math Control Info*, 22(3):257–265, 2005.

[23] J.K. Tar. Towards replacing Lyapunov's 'direct' method in adaptive control of nonlinear systems (invited plenary lecture). *In Proc. of the Mathematical Methods in Engineering Intl. Symp. (MME), Coimbra, Portugal*, 2010.

[24] J.K. Tar, L. Nádai, I.J. Rudas, and T.A. Várkonyi. RFPT-based adaptive control stabilized by fuzzy parameter tuning. *In Proc. of the* 9^{th} *European Workshop on Advanced Control and Diagnosis (ACD 2011), Budapest, Hungary*, pages 1–8, 2011.

[25] T.A. Várkonyi, J.K. Tar, I.J. Rudas, and I. Krómer. Vs-type stabilization of mrac controllers using robust fixed point transformations. *In Proc. of the* 7^{th} *IEEE Intl. Symp. on Applied Computational Intelligence and Informatics, Timişoara, Romania*, pages 389–394, 2012.

[26] K. Kósi, Sz. Hajdu, J.F. Bitó, and J.K. Tar. Chaos formation and reduction in Robust Fixed Point Transformations based adaptive control. *In Proc. of the 4th IEEE Intl. Conf. on Nonlinear Science and Complexity (NSC 2012), Budapest, Hungary*, pages 211–216, 2012.

[27] K. Kósi, Á. Breier, and J.K. Tar. Chaos patterns in a 3 degree of freedom control with robust fixed point transformation. *In Proc. of the* 13^{th} *IEEE Intl. Symp. on Computational Intelligence and Informatics, Budapest, Hungary*, pages 1–5, 2012.

[28] T.A. Várkonyi, J.K. Tar, and I.J. Rudas. Robust Fixed Point Transformations in chaos synchroniztion. *In Proc. of the* 11^{th} *Intl. Symp. of Hungarian Researchers on Computational Intelligence and Informatics, Budapest*, pages 219–224, 2010.

[29] J.K. Tar, L. Nádai, I.J. Rudas, and T.A. Várkonyi. Adaptive emission control of freeway traffic using quasi-stationary solutions of an approximate hydrodynamic model. *Journal of Applied Nonlinear Dynamics*, 1(1):29–50, 2012.

[30] J.L. Chen and Wei-Der Chang. Feedback linearization control of a two-link robot using a Multi-Crossover Genetic Algorithm. *Expert Systems with Applications*, 2(2 Part 2):4154–4159, 2009.

[31] J.K. Tar, J.F. Bitó, I.J. Rudas, and K. Eredics. Comparative analysis of a traditional and a novel approach to Model Reference Adaptive Control. 11^{th} *Intl. Symp. of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary*, pages 93–98, 2010.

[32] J.K. Tar, I.J. Rudas, J.F. Bitó, and K. Kósi. Iterative adaptive control of a three degrees-of-freedom aeroelastic wing model. *Applied Mechanics and Materials*, 300-3001:1593–1599, 2012.

[33] J.K. Tar, I.J. Rudas, J.F. Bitó, and K. Kósi. Robust fixed point transformations in the model reference adaptive control of a three dof aeroelastic wing. *Applied Mechanics and Materials*, 300-3001:1505–1512, 2013.

[34] A. Dineva. *"Adaptive Control of Nonlinear Dynamic Systems with Robust Fixed Point Transformation"– M.Sc. Thesis (in Hungarian).* Óbuda University, Bánki Donát Faculty of Mechanical and Safety Engineering, Budapest. Supervisor: J.K. Tar, 2013.

[35] E.H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9):394–395, 1920.

[36] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[37] S. Chiaverini, O. Egeland, and R.K. Kanestrom. Achieving user-defined accuracy with damped least squares inverse kinematics. *1991 IEEE International Conference on Robotics and Automation, June 19-22, 1991, Pisa, Italy*, page 672, 1991.

[38] J.K. Tar, I.J. Rudas, Gy. Hermann, J.F. Bitó, and J.A. Tenreiro Machado. On the robustness of the Slotine–Li and the FPT/SVD-based adaptive controllers. *WSEAS Transactions on Systems and Control*, 9(3):686–700, 2008.

[39] J.K. Tar, J.F. Bitó, I.J. Rudas, S. Preitl, and R.-E. Precup. Dynamic friction compensation in the Slotine–Li and in an SVD–based adaptive control. *In Proc. of the 17th International Workshop on Robotics in Alpe–Adria–Danube Region (RAAD 2008), 15-17 September, 2008, Ancona, Italy*, CD issue(Paper No. 5), 2008.

[40] G.H. Golub and W. Kahan. Calculating the singular values and pseudoinverse of a matrix. *SIAM Journal on Numerical Analysis*, 2:205–224, 1965.

[41] J.K. Tar, J.F. Bitó, L. Nádai, and J.A. Tenreiro Machado. Robust Fixed Point Transformations in adaptive control using local basin of attraction. *Acta Polytechnica Hungarica*, 6(1):21–37, 2009.

[42] 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.*, 3:133–181, 1922.

[43] K. Kósi, J.K. Tar, and I.J. Rudas. Improvement of the stability of RFPT-based adaptive controllers by observing "Precursor Oscillations". *In Proc. of the* 9^{th} *IEEE Intl. Conf. on Computational Cybernetics, Tihany, Hungary*, pages 267–272, 2013.

[44] 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*, 7(3):65–80, 1927.

[45] Jean-Jacques E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1991.

[46] J.P. Gram. Über die Entwickelung reeler Funktionen in Reihen mittelst der Methode der kleinsten Quadrate. *Journal für die reine und angewandte Mathematik*, 94:71–73, 1883.

[47] E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener. *Matematische Annalen*, 63:442, 1907.

[48] J.K. Tar. Application of local deformations in adaptive control - a comparative survey. In *IEEE 7th International Conference on Computational Cybernetics: ICCC*

2009, Palma de Mallorca, Spain, 2009.11.26-2009.11.29., Budapest: IEEE Hungary Section, pages 25–38., 2009.

[49] B. Csanádi, J.K. Tar, and J.F. Bitó. Matrix inversion-free quasi-differential approach in solving the inverse kinematic task. *In Proc. of the* 17^{th} *IEEE International Symposium on Computational Intelligence and Informatics, 17-19 November 2016, Budapest, Hungary*, pages 61–66, 2016.

[50] József Dombi. A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, 8:149–163, 1982.

[51] J. Aczél. Characterizations of some classes of quasilinear functions with applications to triangular norms and to synthesizing judgements. *Methods Oper. Res.*, 48:3–22, 1984.

[52] B. Kurtán. *Investigation and Adaptive Synchronization of Physiological Neuron Models via Numerical Simulations (MSc Thesis), Supervisor: J.K. Tar*. John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary, 2014.

[53] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophys J.*, 1(6):445–466, 1961.

[54] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proc. of IRE*, 50:2061–2070, 1962.

[55] Brook Taylor. Methodus incrementorum directa & inversa. *Londini: typis Pearsnianis*, 1715.

[56] G. Shilov. Linear Algebra (Dover Books on Mathematics). *Dover Publications*, 1977.

[57] J. Dieudonné. Oeuvres de Camille Jordan I-IV. *Gauthier-Villars, Paris,*, 1961-1964.

[58] J. Somló, B. Lantos, and P.T. Cát. *Advanced Robot Control*. Akadémiai Kiadó, Budapest, 2002.

[59] R. Bupp, D. Bernstein, and V. Coppola. A benchmark problem for nonlinear control design: Problem statement, experiment tested and passive nonlinear compensation. *Proc. of the American Control Conference, Seattle, US*, pages 4363–4376, 1995.

[60] M. Jankovic, D. Fontanie, and P. Kokotovic. Tora example: Cascade and passivity based control design. *IEEE Trans. on Control Systems Technology*, 4:292–297, 1996.

[61] P. Baranyi, Z. Petres.P. Várlaki, and P. Michelberger. Observer and control law design to the tora system via tpdc framework. *WSEAS Transctions on Systems*, 1(5):156–163, 2006.

[62] D. S. Bernstein(ed.). A special issue presenting 9 papers on controlling the 'tora' system. *International Journal of Robust and Nonlinear Control*, 8:305–457, 1998.

[63] L. Főző, R. Andoga, and L. Madarász. *Mathematical model of a small Turbojet Engine MPM-20, In I.J. Rudas, J. Fodor, J. Kacprzyk (eds.) Studies in Computational Intelligence Vol. 313, Computational Intelligence in Engineering pp. 313-322*. Springer-Verlag Berlin Heidelberg, 2010.

[64] R. Andoga, L. Főző, L. Madarász, J. Považan, and J. Judičák. Basic approaches in adaptive control system design for small turbo-compressor engines. *In: Proc. of the IEEE 18th International Conference on Intelligent Engineering Systems (INES 2014), from 3 to 5 July, 2014, Tihany, Hungary*, pages 95–99, 2014.

[65] R. Andoga, L. Főző, L. Madarász, and T. Karol'. A digital diagnostic system for a small turbojet engine. *Acta Polytechnica Hungarica*, 10(4):45–58, 2013.

[66] R. Andoga, L. Madarász, L. Főző, T. Lazar, and V. Gašpar. Innovative approaches in modeling, control and diagnostics of small turbojet engines. *Acta Polytechnica Hungarica*, 10(5):81–99, 2013.

[67] A. Jadbabaie. *Receding Horizon Control of Nonlinear Systems: A Control Lyapunov Function Approach*. PhD Thesis, California Institute of Technology, Pasadena, California, USA
http://www.seas.upenn.edu/~jadbabai/papers/Phdthesis.pdf, Visited: 2014. Jan. 15., 2000.

[68] A. Jadbabaie, J. Yu, and J. Hauser. Stabilizing receding horizon control of nonlinear systems: A control Lyapunov function approach. *In the Proc. of the American Control Conference, 1999*, pages 1535–1539, 1999.

[69] A. Jadbabaie, J. Yu, and J. Hauser. Receding horizon control of the Caltech ducted fan: A control Lyapunov function approach. *In the Proc. of the IEEE Conference on Control Applications, 1999*, pages 51–56, 1999.

[70] J. Hauser and C.C. Chung. Nonlinear control of a swinging pendulum. *Automatica (Journal of IFAC)*, 31(6):851–862, 1995.

[71] E. I. Butikov. On the dynamic stabilization of an inverted pendulum. *American J. of Physics*, 69:p.755, 2001.

[72] P. L. Kapitza. Dynamic stability of a pendulum when its point of suspension vibrates. *Soviet Phys. JETP*, 21:588–592, 1951.

[73] A. Stephenson. Xx. on induced stability. *Philosophical Magazine Series 6*, 15(86):233–236, 1908.

[74] J. Aracil, F. Gordillo, and J.A. Acosta. Stabilization of oscillations in the inverted pendulum. *In Proc. of the: 15 IFAC World Congress*, page p.261, 2002.

[75] R.E. Belman, J. Bentsman, and S. M. Meerkov. Vibrational control of a class of nonlinear systems: Vibrational stabilization. *IEEE Trans. Aut. Control*, 32(8):710716, 1986.

[76] H. Yabuno, K. Goto, and N. Aoshima. Swing-up and stabilization of an underactuated manipulator without state feedback of free joint. *IEEE trans. Robotics and Automation*, 20:259365, 2004.

[77] H. Yabuno. and K. Tsumoto. Experimental investigation of a buckled beam under high-frequency excitation. *Archive of Applied Mechanics*, 77(5), 2007.

[78] I.V. Miroshnik and N. Odinets. Stabilization of pendulum oscillations around upper position. *In: 6th IFAC Syposium on Nonlinear Control Systems (NOLCOS 2004)*, 3, 2004.

[79] V. Piuri and S. Ablameyko. Introduction to neural networks for instrumentation, measurement, and industrial applications. *Neural networks for instrumentation, measurement and related industrial applications, S. Ablameyko, L. Goras, M. Gori, V. Piuri (Eds.), IOS Press, ISBN: 1-586-03303-4*, 2003.

[80] Patrick Wong, Fred Aminzadeh, and Masoud Nikravesh, editors. *Soft Computing for Reservoir Characterization and Modeling*. Springer- Verlag Berlin Heidelberg, 2002.

[81] Hirosato Seki. Nonlinear function approximation using fuzzy functional sirms inference model. *AIKED'12 Proceedings of the 11th WSEAS international conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 201–206.

[82] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3):199–249.

[83] I. Daubechies. The wavelet transform, time frequency localizlocal and signal analysis. *IEEE Trans on Inf. Theory*, 36:961–1005, 1990.

[84] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of l^2. *Trans. Am. Math. Soc.*, (315):69–78, 1989.

[85] F. Marino, V. Piuri, and E.Jr. Swartzlander. A parallel implementation of the 2-d discrete wavelet transform without interprocessor communications. *IEEE Transactions on Signal Processing (ISSN: 1053-587X)*, pages 3179–3184, November 1999.

[86] D.L. Donoho. Denoising by soft-thresholding. *IEEE Trans on Inf. Theory*, 41(3):613–627, 1995.

[87] S. Zilberstein. Operational rationality through compilation of anytime algorithms. *PhD Thesis*, 1993.

[88] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.

[89] A. R. Várkonyi-Kóczy and A. Rövid. Fuzzy logic supported frequency range estimation for adaptive fourier analysis. In *SCIS and ISIS 2006 Proc. of the Joint 3rd Int. Conf. Soft Computing and Int. Syst. and 7th Int. Symp. on Advanced Intelligent Systems, Tokyo, Japan, 2006.09.20-2006.09.24., Paper 100432*, 2006.

[90] D.L. Donoho and M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the Ammerican Statistical Association*, 90(1200-1224), 1995.

[91] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, 1981.

[92] A. R. Várkonyi-Kóczy. Fuzzy approaches in anytime systems. *In: R Seising, E Trillas, C Moraga, S Termini (Eds.), On Fuzziness.: A Homage to Lotfi A. Zadeh. Berlin: Springer-Verlag*, pages 307–317, 2012.

[93] A. R. Várkonyi-Kóczy. Adaptive anytime data transmission of non-stationary signals. *JOURNAL OF ADVANCED COMPUTATIONAL INTELLIGENCE AND INTELLIGENT INFORMATICS 14:(3)*, pages 240–246, 2010.

[94] P. J.Rousseeuw and A. M. Leroy. Robust regression and outlier detection. *John Wiley & Sons*, 2005.

[95] A. Ivera-Azcarate and et.al. Outlier detection in satellite data using spatial coherence. *Remote Sensing of the Environment*, 118:84–91, 2012.

[96] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the Am. Stat. Ass.*, 74:829–836, 1979.

[97] W. S. Cleveland and R. McGill. The many faces of scatterplot. *Journal of the Am. Stat. Ass*, 79:829–836, 1984.

[98] A. Savitzky and Golay M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627 1639, 1964.

[99] H.H. Madden. Comments of savitzky-golay convolution method for least squares fit smoothing and differentiation of digital data. *Anal. Chem.*, 50:1383–1386, 1978.

[100] J. Steiner, Y. Termonia, and J Deltour. Comments on smoothing and differentiation of data by simplified least squares procedura. *Anal. Chem.*, 4:1906–1909, 1972.

[101] T.H. Edwards and P.D Willson. Digital least squares smoothing of spectra. *Appl. Spectroscopy*, 28:541–545, 1974.

[102] K. Ahnert and M. Abel. Numerical differentiation of experimental data local versus global methods. *Computer Physics Communications*, 177:764–774, 2007.

[103] M. Browne, N. Mayer, and T.R.H. Cutmore. A multiscale filter for adaptive smoothing. *Digital Signal Processing*, 17:69–75, 2007.

[104] J.H. Wayt and K. R. Integrated savitzky-gola filter from inverse taylor series approach. In *Proc. of the 2007 15th Intl. Conf. on Digital Signal Processing (DSP 2007), IEEE*, pages 375–378. IEEE, 2007.

[105] A. Zhao, X. Tang, and Z. Zhang. The parameters optimization selection of savitzky-golay filter and its application in smoothing pretreatment for ftir spectra. In *Proc. of the IEEE 9th Conference on Industrial Electronics and Applications (ICIEA)*, pages 516–521, 2014.

[106] Per-Olof Persson and G. Strang. Smoothing by savitzky-golay and legendre filters. In Gilliam S.D. Rosenthal, J., editor, *Mathematical Systems Theory in Biology, Communications, Computation, and Finance*, volume 134, pages 301–315. Springer New York, 2003.

[107] R.S. Krishnan and C.S. Seelamantula. On the selection of optimum savitzky-golay filters. In *IEEE Trans. on Signal Processing*, volume Vol. 61 (2), pages 380–391, 2013.

[108] J. Engel and et al. Breaking with trend in pre-processing? *Trends in Analytical Chemistry*, 2013:96–106, 2013.

[109] Lukasz Komsta. A comparative study on several algorithms for denoising of thin layer densitograms. *Analytica Chimica Acta*, 641:52–58, 2009.

[110] Cs. Finta, P. J. Teppola, M. Juuti, and P. Toivanen. Feasibility of parallel algorithms and graphics processing unit acceleration in chemical imaging. *Chemometrics and Intelligent Laboratory Systems*, 127:132–138, 2013.

[111] P. Tong, Y. Du, K. Zheng, T. Wu, and J. Wang. Improvement of nir model by fractional order savitzky-golay derivation (fosgd) coupled with wavelength selection. *Chemometrics and Intelligent Laboratory Systems*, 143:40–48, 2015.

[112] R. W. Schafer. What is a savitzky-golay filter? (lecture notes). *Signal Processing Magazine, IEEE*, 28 (4):111–117, 2011.

[113] R. W. Schafer. On the frequency-domain properties of savitzky-golay filter. In *In Proc. of the the 2011 DSP/SPE Workshop, Sedona, AZ,*, pages 54–59, 2011.

[114] M. U. A. Bromba and H. Ziegler. Application hints for savitzky-golay digital smoothing filters. *Anal. Chem.*, 53(11):1583–1586, 1981.

[115] R. W. Hamming. *Digital Filters*. Prentice-Hall, Englewood Cliffs, NJ, Chapter 3, 3rd ed. edition, 1989.

[116] J. Luo, K. Ying, P. He, and J. Bai. Properties of savitzky-golay digital differentiators. *Digital Signal Processing*, 15(2):122–136, 2005.

[117] J. Luo, K. Ying, and J. Bai. Savitzky-golay smoothing and differentiation filter for even number data. *Signal Processing*, 85(7):1429 – 1434, 2005.

[118] D. Chen and et al. Digital fractional order savitzky-golay differentiator. *IEEE Trans. on Circuits and Systems*, 58(11):758–762, 2011.

[119] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley and Sons, 4th ed. edition, 2008.

[120] B.P. Flannery, H.W. Press, A. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 2 edition, 1992.

[121] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proeedings of the 1968 ACM National Conference*, pages 517–524, 1968.

[122] József Dombi. Towards a general class of operators for fuzzy systems. *IEEE Trans. on Fuzzy Systems*, 16(2):477–484, 2008.