ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

PhD Thesis

Application of Novel Approaches in Optimal and Adaptive Optimal Control

Hazem Issa

Supervisor
Prof. Dr. József K. Tar

Doctoral School of Applied Informatics and Applied Mathematics

July 2023

Members of the Complex Examination Committee:

Chairman:	Prof.Dr.habil. Róbert Fullér DSc, Széchenyi University, Győr
Members:	Prof.Dr.habil. László Pokorádi, Óbuda University, Budapest
	Dr. György Eigner, Óbuda University, Budapest

Public defense Committee:

Chairman:	Prof.em.Dr. Aurél Galántai DSc, Óbuda University, Budapest
Secretary:	Dr. Zoltán Léka, Óbuda University, Budapest
Opponents:	Dr. Ákos Odry, University of Szeged, Szeged
	Dr. Krisztián Kósi, Óbuda University, Budapest
Members:	Dr.habil. Balázs Németh, SzTAKI, Budapest
	Dr.habil. Katalin Gambár, Óbuda University, Budapest

Date of Public Defense:

Declaration of Authorship

I, Hazem Issa, declare that the thesis entitled, “Application of Novel Approaches in Optimal and Adaptive Optimal Control” and the content presented in it, is entirely my own research work in the able supervision of Prof. Dr. habil. József K. Tar. D.Sc. It is hereby assured that:

- This work has been completed as a candidate for Ph.D. degree at Óbuda University.
- Wherever in the thesis, the published work of other authors is consulted, this work is clearly referenced.
- Wherever in the thesis, quotations from other authors are used, then this fact is clearly mentioned. The thesis is absolutely my own work with the exception of such quotations.
- All the main sources of help have been properly acknowledged.

Budapest, 5 July, 2023.
Hazem Issa

Acknowledgment

I would like to express my deepest gratitude and appreciation to all those who have contributed to the completion of this thesis.

First and foremost, I am sincerely grateful to my respected supervisor Prof.Dr.habil.József K. Tar D.Sc for his patience, genuine support and professional guidance in both personal and scientific levels. His expertise, insights, and constructive feedback have been instrumental in shaping the direction and quality of this work.

I extend my heartfelt thanks to the members of examination and the defense committee, for their time, expertise, and valuable input. Their thoughtful suggestions and critical evaluation have significantly contributed to the refinement of this work.

Special thanks to Ms. Zsuzsanna Bácskai for her huge support and helping me for organizing the dissertation process.

I gratefully acknowledge the support of the Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University for providing a conducive academic environment and necessary resources for carrying out this research. I am also thankful to the Tempus Public Foundation for providing the Stipendium Hungaricum program.

Last but not least, I would like to express my deepest appreciation to my family for their unconditional love, encouragement, and unwavering belief in my abilities. Their constant support, patience, and understanding have been the pillars of strength that enabled me to pursue this academic endeavor.

I deeply thankful to everyone who has contributed, directly or indirectly, to the completion of this thesis.

Hazem Issa

Contents

Acknowledgment **iii**

Abstract **1**

1 Introduction **2**

1.1 Motivations and Goals 2

1.2 State of Art 3

1.3 Research Chapters and their Related Aims 7

1.4 Research Methodology 8

2 Investigations About Computational Acceleration For Optimization Problems **9**

2.1 Complexity Reduction by using the Gram-Schmidt Algorithm 9

2.1.1 A Simple Numerical Example 10

2.2 Solving the Inverse Kinematic Task by Accelerated Reduced Gradient Algorithm 14

2.2.1 Definition of the Kinematic Parameters and the Differential Inverse Kinematic Task 15

2.2.2 Setting the Cost Function and the Simulation Parameters 17

2.2.3 Simulation Results 18

2.3 Thesis Statement I 22

3 Improvement of the Fixed Point Iteration-based Adaptive Receding Horizon Controller **24**

3.1 The Fixed Point Iteration-based Adaptive Control 24

3.1.1 The Response Function 25

3.1.2 Kinematic Requirements 25

3.1.3 Convergence Issues 26

3.1.4 On the Content of Block Deformation 27

3.2 The Receding Horizon Scheme-based Approach without Gradient Reduction 31

3.3 Further Modifications in the Original RHC Approach 32

3.3.1 Simulation Examples for a Furuta Pendulum 33

3.3.1.1 Simulations with LuGre Friction Model 38

3.3.2 Design of Adaptive Receding Horizon for SCARA Robot 40

3.3.2.1 Simulation Results 43

3.4 General Receding Horizon Solution of the Inverse Kinematic Task 49

3.4.1 Simulation Results 51

3.5 Tackling Actuator Saturation in Fixed Point Iteration-based Adaptive Control 55

3.5.1 Simulation Results 56

3.5.1.1 Simulations with LuGre Friction Model 62

3.6 Thesis Statement II 65

4 Investigation of the Cooperation of Noise Filtering Methods With Fixed Point Iteration-based Adaptive Techniques 67

4.1 Using High Frequency Noisy Signal Filtering 67

4.2 Application of Simple Moving Window with Affine Signal Approximation . 68

4.3 Continuous Time Fixed Point Iteration-based Control 68

4.4 Performance of Fixed Point Iteration-based Adaptive Control in Noise Filtering 70

4.5 Comparison of Noise Filtering between Fixed Point Iteration-based Adaptive Control and Unscented Kalman Filter 76

4.5.1 The Compared Noise Filtering Methods 76

4.5.1.1 General Assumptions Regarding the Inherited Noise Distribution in UKF 79

4.5.2 Simulation Results 80

4.5.2.1 Comparisons Without Noise 80

4.5.2.2 Comparison With Gaussian Distribution Noise 82

4.5.2.3 Comparison With Logistic Distribution Noise 86

4.6 Thesis Statement III 88

5 Implementing Fixed Point Iteration-based Adaptive Control and Particle Swarm Optimization 89

5.1 Limitations in the Usage of Fixed Point Iteration-based Adaptive Control and Particle Swarm Optimization Illustrated by a Simple Paradigm 90

5.1.1 The Dynamic Model of the van der Pol Oscillator 90

5.1.2 Computed Simulations 91

5.1.2.1 Simulations for Sinusoidal Motion 92

5.1.2.2 Computations for Increasing Frequency and Amplitude . . 96

5.2 Application for a Realistic Robot Model 100

5.2.1 Model Dynamics 102

5.2.2 Implementation of the PSO Strategy 102

5.2.3 Simulation Results 103

5.3 Thesis Statement IV 110

6 Conclusions 112

7 Future Research 113

Appendix A 115

Own Publications 117

References 128

List of Acronyms

ARHC: Adaptive Receding Horizon Controller
CTC: Computed Torque Control
DOF: Degree of Freedom
FPI: Fixed Point Iteration
MPC: Model Predictive Control
PSO: Particle Swarm Optimization
RHC: Receding Horizon Controller
TCP: Tool Center Point
UKF: Unscented Kalman Filter

Abstract

In my Thesis I elaborated certain improvements in the subject areas of Optimal and Adaptive controllers with the main aim of realizing their efficient integration. In the traditional mainstream, optimal controllers are based on the mathematical foundations of functional optimization under constraints. The adaptive controllers that tackle strongly nonlinear problems normally use Lyapunov functions for the calculation of the control signal. Both structures have their own inner rigidity that makes their combination not trivial. Recognizing that the mathematical structure of the alternative of the Lyapunov function-based technique, the Fixed Point Iteration-based adaptive control immediately allows the integration of the various particular variants of these methods, I concentrated on the elaboration of adaptive optimal controllers. I have shown that by eliminating the constraint terms in the optimal control by incorporating the dynamic model in the cost functions the computational burden of the method can be significantly reduced in the case of quite sophisticated cost functions. I also elaborated improvement in the FPI-based adaptive controllers by reducing its noise sensitivity using simple filtering techniques, and efficiently tackled the problem of actuator saturation. I suggested separate or optionally partly coupled application of this adaptive method with parameter identification purposes the need of which arose not from the side of control applications. I pointed out the limitation of the evolutionary algorithms in parameter identification issues, and suggested the use of the improved identified model with the same adaptive technique to improve its stability. The statements of my Thesis are underpinned by simulation investigations I made by the use of Julia language programs.

Chapter 1

Introduction

1.1 Motivations and Goals

In practical control tasks various typical problems must be tackled as e.g., the lack of precise dynamic model of the system to be controlled, the presence of unknown external disturbances and their consequences, limited physical possibilities for introducing control force or other equivalent action into the controlled process, limited possibilities for measuring or at least estimating the controlled system's actual physical state. For tackling at least certain elements of these problems various control methodologies have been elaborated. Some of them were based on particular mathematical formalism that do not seem to be easily combined with each other. In this field of efforts there is a plenty of open problems and a general lack of integration can be observed. The question naturally arose: is it possible to reach certain achievement in this direction? My work was motivated by this simple question. Explanation of the precise mathematical details of which I obtained my motivations can be given only after the state-of-the-art review. In the sequel I give only a short summary of my ideas.

The Model Predictive Controller (MPC) was early idea developed for utilizing the available dynamic model of the controlled system. In order to provide the designer possibility for force limitations or complying with various other, often contradictory requirements, these controllers mathematically were formulated as solutions to the optimization under constraints tasks. Normally, the cost functions represented some weighted sum of various penalty terms that should have been minimized, while the dynamic model of the controlled system appeared in the constraint terms. In the early dynamic programming approach the mathematical background was the minimization of functionals with strong analogy of the variation principles of Classical Mechanics. The computational power -requirements of this approach was considerably reduced by the application of time grid with discrete time-resolution in the Receding Horizon Controller (RHC) that maintained the constraint terms with combination with the Lagrange multipliers. The practical applicability of this approach strongly depended on the dynamics (speed) of the physical process to be controlled, and the technological level of the available processors that were able to realize these control methods. **I realized that in this formalism there is a great freedom for the designer to play with the possibilities for distributing certain details between the cost and the constraint terms. This possibility made me carrying out simulation investigations.**

Whenever the computational resources were not rich enough to apply the complex mathematical framework of optimized controllers, as e.g., in Robotics in the nineties of the past century, the dynamic model was utilized without being placed into the mathematical framework of optimal controllers. The so called Computed Torque Control (CTC) had to scope with the problem of lacking reliable and precise dynamic models. It became clear in the early nineties that no such models can be constructed even within the Classical Mechanics-based framework that does not consider friction effects. In the literature coping with the problems of incomplete and imprecise models generally happens by the use of either Robust design as the Variable Structure / Sliding Mode Controller, or by the application of Adaptive techniques. The Sliding Mode controller often produces dangerous excitation of the controlled system while the mainstream of the finer adaptive controllers is based on Lyapunov's 2^{nd} or direct method. **I recognized that from various points of view this method is very complicated, needs complete state estimation, so the question naturally arose whether is it possible to combine with the optimal control approach its possible, more simple alternative that was based on Banach's fixed point theorem? For this reason I carried out simulation investigations with this mathematically simple approach.**

In the modern control approaches the efficient tool of evolutionary algorithms-based system identification methods can be found. It was a natural idea to investigate how is it possible to combine evolutionary methods-based system identification techniques with the fixed point iteration-based adaptive control.

In dynamic control problems the presence of observation or sensor noise is a general problem. Due to the structure of the fixed point iteration-based approach it was expected that the noise issues have enhanced significance in this case. Therefore, I felt strong motivation to consider noise filtering possibilities. The success of the Acceleration Feedback Controllers that have to cope with similar problems convinced me that it is not hopeless to make investigations in this direction.

1.2 State of Art

In control technology, various control methods are present in the inventory of possible solutions. The appropriate choice can be selected according to various particular and practical aspects related to a given task, and there is no way to generally state that a given method would be superior in comparison with others. Such properties as mathematical complexity, the need for computational power, the need for a more or less precise dynamic model of the controlled system, robustness against modeling errors and external disturbances, adaptivity, and possibilities for implementation can be considered when a control approach is chosen to tackle a given problem.

The wide set of model-based controllers belong the (MPC) (e.g., [1]) in which beside the dynamic model of the controlled system various force limitations and other restrictive factors can be taken into account that originate from sources other than the model. It was successful especially in the control of slow processes as e.g., crystallization, that can be traced by computers of limited computational power. For finding the optimum a computationally greedy approach was suggested in the late Fifties of the past century as Dynamic Programming that aimed at the minimization of functionals [2, 3]. The computational power of this approach later was reduced by solving this problem over a discretized time-grid and using numerical approximations under the name Receding Horizon Controller [4]. However, the original elegant structure in general was maintained. The analogy with the Canonical Equations of Motion of Classical Mechanics were maintained with related issues as the flow of incompressible fluids and conservation rule for the Hamiltonian (or artificial Hamiltonian) of the controlled system (e.g., [5]). These controllers generally applied Lagrange's Reduced Gradient Algorithm [6]for problem solution. In special cases as linear time-invariant system models and quadratic cost functions, also utilizing the theoretical results by Riccati [7] and Shur [8, 9], considerable mathematical simplifications were achieved under the name Linear-Quadratic Regulators [10]. This approach was able to treat modeling errors in a very simple manner: the optimized motion was designed for a whole horizon, but normally only one step of this optimized force was exerted on the system, and in the next step a new horizon have been calculated again.

Whenever the computational resources were not rich enough to apply the complex mathematical framework of optimized controllers, as e.g., in Robotics in the nineties of the past century, the dynamic model was utilized without being placed into the mathematical framework of optimal controllers. The so called Computed Torque Control (CTC) [11] had to scope with the problem of lacking reliable and precise dynamic models. It became clear in the early nineties that no such models can be constructed even within the Classical Mechanics-based framework that does not consider friction effects [12]. Various friction models were considered with regard to slow velocity motion where the phenomenon stick–slip' typically occurs (e.g., [13–15]). It was also realized that building in even the simplest friction models leads to observation or identification problems (e.g., [16]). High complexity friction models need the introduction of complementary degree of freedom (DOF) as in the LuGre model [17].

In many cases, a simple PID-type controller invented in the 1940s [18] can do well. In robotics, the direct use of the dynamic model without inserting it into the mathematical framework of optimal controllers was initiated in the 1980s [11] in the concept of (CTC). In this approach the inverse dynamic model is directly used for the calculation of the control signal without using the mathematical complexity of the optimal controllers. However, it became clear very early that practically it is impossible to develop precise dynamic robot models (e.g., [12]), and that the identification of important parameters related to modeling the friction effects has limitations, too (e.g., [16]).

The robust variable structure/sliding mode controllers that became popular in the 1990s (e.g., [19]) are simple solutions that can solve the problem of modeling errors and unknown external disturbance. In a similar manner, resolved acceleration rate control (e.g., [20]) and acceleration feedback controllers (e.g., [21]) can be considered as improvements of the CTC controllers.

The wide class of adaptive controllers tackle the imprecisions of the models in different manners. The widest subset uses Lyapunov's stability theorems and keeps prevailing from the early 1990s to the present (e.g., [22]) as well as the model reference adaptive control (e.g., [23]). Normally, stability or asymptotic stability of these solutions are guaranteed for a huge set of possible parameters of which the appropriate ones can be selected on the basis of practical aspects often applying various versions of evolutionary computation as genetic algorithms [24, 25], particle swarm optimization [26–29], simulated annealing [30], and so on.

Strong non-linearity is a natural feature of most physical, biological, economic, and engineering systems. In spite of that most of traditional software packages solving optimization problems can normally handle only linear time-invariant system models with typically quadratic cost function structures because this restricted subject area can be tackled by well-elaborated and efficient mathematical tools as the application of Riccati equations [7] (it provides the solution of special first-order quadratic differential equations by solving second-order linear ones), Schur's decomposition method that obtains the solution of quadratic matrix equations by solving linear ones [8, 9]. For solving linear matrix inequalities in system and control theory a complete program was announced by Boyd et al. in 1994 [31] for which efficient MATLAB program packages have been developed [32]. The mainstream of the engineering research efforts aimed at the elaboration of approximate linear system models and quadratic cost functions for tackling optimization problems by the use of this efficient mathematical apparatus. In [33], the linear matrix inequality (LMI) condition based on slack variables was used to reduce the high gains of control, resulting in using the robust $\mathscr{H}_\infty$ state feedback controllers.

However, for more complex dynamical models and specially structured cost functions the more general mathematical context does not allow such relatively simple solutions. Instead of using ready-made program packages researchers have to develop their own program codes that are not supported by the rigorous and reliable quality guaranties of the MATLAB packages.

From a mathematical point of view, optimization can be formulated by the use of variation calculus. In the 1950s, i.e., in the advent of the appearance of powerful computers, Bellman introduced dynamic programming [2] that computationally is too greedy. The problem was later simplified by the introduction of a discrete, evenly scaled time-grid of resolution δt that is dense enough to allow numerical differentiation and Euler integration over it. The sum of the cost function contributions in the grid points of a horizon of discrete length H was minimized for a first-order dynamical system under the constraint $\frac{q(t_{i+1})-q(t_i)}{\delta t} \approx \dot{q}(t_i)$ in which the function $\dot{q}(t_i) = \mathscr{F}(q(t_i), u(t_i))$ describes the dynamic model of the controlled system, and $u(t_i)$ denotes the control force. By the use of the usual constraint function $g_i(q(t_i), q(t_{i+1}), u(t_i)) := \frac{q(t_{i+1})-q(t_i)}{\delta t} - \mathscr{F}(q(t_i), u(t_i))$, a general cost function (with a simpler notation) $\sum_{\ell=1}^{H} \Phi(q_\ell, u_\ell)$ has to be minimized over the horizon by varying the coordinates $\{q_2, \dots, q_H\}$ (q_1 is given as the initial condition of the motion), and force terms $\{u_1, \dots, u_{H-1}\}$ (because u_H has influence only on the next grid point at time t_{H+1}). The optimization must have done under the constraints $g_i(q_i, q_{i+1}, u_i) = 0$. It traditionally can be solved by the use of Lagrange's reduced gradient method by using Lagrange multipliers for gradient reduction that was introduced in the late 18th century for solving constrained problems in Classical Mechanics [6]. It can well be used for controlling slow processes as e.g., crystallization in chemistry [1, 34] and traffic control [35, 36]. Later, it obtained ample applications from the 1960s with the development of computer technology that provided easy implementation possibilities (e.g., [37, 38]). The scheme description is known as the (RHC) [4] which is a reliable, heuristic realization of the (MPC) (e.g., [39, 40]) that has many applications (e.g., [41–46]). The adaptive version of RHC were investigated in many cases, such as in [47] wherein the used Adaptive Receding Horizon Controller (ARHC) is based on Lyapunov's adaptation law, whereas in [48], the adaptive controller is based on the set membership identification algorithm, which iteratively calculates at each cycle a set of candidate plant models. The general ARHC is used in [49] along with particle swarm optimization (PSO). Implementing the sliding mode (SM) as an adaptive technique for ARHC is addressed in [50].

Because of the fact that the Lagrange multipliers normally have clear physical interpretation (e.g., [51]), and the strong analogy with the canonical equations of Classical Mechanics [5, 52, 53] that provides solutions similar to the flow of incompressible fluids, together with the plausible mathematical consequences of this approach, the constraint-based formulation of the problem generally prevailed, though it is not the computationally simplest and cheapest approach. These analogies are derived from considering the auxiliary function of the problem in (1.1)

$$\Psi(\{q\},\{\lambda\},\{u\}) := \sum_{\ell=1}^{H} \Phi(q_\ell,u_\ell) - \sum_{\ell=1}^{H-1} \lambda_\ell g_\ell(q_\ell,q_{\ell+1},u_\ell) \quad . \tag{1.1}$$

Evidently, $\Psi(\{q\},\{\lambda\},\{u\})$ is not bounded, and at the point where the gradient reduction algorithm stops, it satisfies the equations as $\frac{\partial\Psi}{\partial\lambda_j} = 0$, meaning that the solution satisfies the constraint conditions, $\frac{\partial\Psi}{\partial q_k} = 0$ that can be so interpreted that the reduced gradient is 0, and an additional condition $\frac{\partial\Psi}{\partial u_i} = 0$. These partial derivatives allow the interpretation of the appearance of the numerical approximation of a differential equation for $\dot{\lambda}$, considering the q_i and λ_i pairs as canonical coordinate pairs, and interpreting Ψ as a Hamiltonian with the conservation property $\dot{\Psi} \equiv 0$. The analogy with the flow of incompressible fluids is related to the fact that the canonical state propagation equations are related to symplectic transformations that conserve the volume of the phase space (Liouville's theorem, e.g., [5]).

The numerical algorithm that solves the above problem is commenced by finding a point on the constraint surface by using the Newton–Raphson algorithm [54–56], then making consecutive small steps along the reduced gradient $\nabla\Phi - \sum_\ell \lambda_\ell \nabla g_\ell$ in which the Lagrange multipliers are so chosen that for the constraint gradients it must be valid that $\forall j$ $\left(\nabla g_j\right)^T (\nabla\Phi - \sum_\ell \lambda_\ell \nabla g_\ell) = 0$ (in this formulation the symbol ∇ contains $\frac{\partial}{\partial q}$ and $\frac{\partial}{\partial u}$ components). Gradient reduction needs the solution of this linear set of equations. The algorithm stops when the reduced gradient becomes zero. It was realized that placing the dynamic model into the constraint term of the optimization task is rather a tradition than a necessity. If we do not insist on the above mentioned elegant formal analogies with classical mechanics, the complexity of the calculations can be considerably reduced. In the original approach the free variables of the optimization are the coordinate values $\{q\}$, and the force terms $\{F\}$ over the horizon, and the quantities that additionally have to be calculated are the $\{\lambda\}$ Lagrange multipliers for reduction of the gradient containing the partial derivatives according to the components $\{q\}$ and $\{F\}$. In [57], the structure of the auxiliary function was investigated in the case of a simple paradigm, and it was found that the appropriate solution is at its saddle point. Furthermore, instead of using a set of individual constraint functions for optimization as $\{g_\ell = 0\}$, the use of a single constraint term defined as $G := \sum_\ell g_\ell^2 = 0$ can be successfully applied with only one associated Lagrange multiplier that can very easily be computed. In [58], the use of the Lagrange multipliers was completely evaded, and the method's operation was illustrated by controlling the dynamic model of two connected mass points that were able to move in a given linear direction. In this approach, the free variables of the optimization are only the force terms $\{F\}$ over the horizon, the gradient in the optimization consists only of the $\frac{\partial}{\partial \mathrm{F}}$ components, and the simple gradient descent method can be applied without any gradient reduction. Following this simple illustration, the method was used for simulating the treatment of illness type 1 diabetes mellitus in determining the necessary insulin ingress rate, and the estimation of the evolution of the not observable internal model variables. In [59], this approach was considered for the RHC control of the Furuta pendulum [60], and in [61, 62] application possibilities were considered in solving the inverse kinematic task of redundant robots.

With regard to the significance of the effects of measurement noises, in the traditional literature, in which normally PID-type feedback terms are used, and in the adaptive control some Lyapunov function-based techniques are prevailing, the noise terms are modeled as additional terms of more or less marginal significance. Generally it is assumed that the physical causes of the noise terms are not identified and individually modeled. It is assumed that the effect of a large number of statistically independent noise sources produce some result of normal distribution with zero mean if the appropriate sensors are well installed. The effect of the lost signals together with the additional Gaussian noise are assumed and, for instance, the Kalman filters are so designed that they are optimized for this Gaussian spectrum (e.g., [10, 63–65]). However, it can be emphasized that the digital components in the realistic applications do not allow to realize the long tail of the Gaussian distributions, and the originally causal models are treated as really causal ones burdened with the additional noise terms. (The control of stochastic processes is absolutely out of the scope of my dissertation.) Since the Fixed Point Iteration-based method feeds back higher order derivatives than the traditional ones, and the adaptation mechanism of this approach learns from the observations of the recent time instances, any noise filtering technique causes some delay that may corrupt this very special and primitive learning method. Therefor, the use of this method made it necessary to consider noise filtering issues.

Since in the numerical control the mathematical properties of the differentiation can cause high noise-like contribution, the traditional approach simply applies low pass filters as e.g., Bodó and Lantos in [66], in my dissertation I also applied it in one of the theses. For fixed point iteration-based control specific preliminaries as ad hoc ideas can be found in [67–70].

1.3 Research Chapters and their Related Aims

This Thesis consists of seven chapters where four of which are the main research ones. Each of them has related particular studies that are organized as follows: Chapter 2 will investigate the possibility of reducing the complexity of solving constrained optimization problems by using the Gram-Schmidt Algorithm. While at the second part, the suggested alternative method will be tested by using a 7-DOF redundant robot system. Chapter 3, the modification of the Receding Horizon Controller will be suggested, in which the use of the constraint terms is completely evaded, and the dynamic model is built in the cost function by the use of little redundancies. Next, the applicability and limitations of this modification will be investigated by using the dynamic control (Furuta pendulum and a 4-DOF SCARA robot model) , and, on the basis of some formal analogy, in the solution of the differential inverse kinematic tasks of redundant robot arms. Also, the actuator saturation and windup effects will be introduced. Chapter 4 will study the cooperation between the FPI-based adaptive technique with the simple noise filtering techniques. Further, these combinations will be compared against Unscented Kalman Filter. Following, Chapter 5, will probe the limitations of the applicability of Particle Swarm Optimization (PSO) with Fixed Point Operation-based adaptive controllers in on-line and off-line modes. Finally, the conclusions of the Thesis will be outlined in Chapter 6 while the future research possibilities will be proposed in Chapter 7.

1.4 Research Methodology

All investigations of the proposed theoretical algorithms and their engineering applications are tested via running the simulation on Julia Language. The reason behind selecting Julia is related to its wide availability as a free and open-source software, capability for providing professional visualization methods that fit the publication requirements. Besides that, its fast processing that reflects the most modern computational technologies against the lack of resources was an important motivation for its use.

The used editions of Julia during the research are v1.4.2-win64 (May 25, 2020), v1.4.2-win64 (July 7, 2021) and lastly v1.8.0 (September 6, 2022). The supporting environments are Python v3.7 and v3.9 so that the visualization in Julia and the possible use of the *qt* back-end (i.e., it applies the Python's Matplot Library) were important facts that motivated my choice.

Chapter 2

Investigations About Computational Acceleration For Optimization Problems

The Lagrange's method has considerable computational needs that mathematically describe the constrained physical systems in Analytical Mechanics [6]. However, nowadays, in the possession of cheap computers and software products it became a practical problem solving tool for instance in the Solver package of MS EXCEL that can be applied in financial and technical problems (e.g., [71, 72]). In many cases the Lagrange multipliers have important physical meaning, therefore it is necessary to compute them (e.g., [51, 73]), and the Solver package also computes them.

This chapter, in its first part, investigates the possibility of speeding up the computations of solving constrained optimization problems by avoiding the calculations of individual Lagrange multipliers. While at the second part, the suggested alternative method is tested by using a 7-DOF redundant robot system.

2.1 Complexity Reduction by using the Gram-Schmidt Algorithm

Consider the general form of the task Optimization Under Constrains: Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ a differentiable cost function and set the problem as follows:

$$\begin{aligned}&\text{Find min. of } f(x) \text{ under the constraints that}\\ &\forall k \in \{1,\dots,K\} \;\; g_k(x) = 0 \;\; ,\end{aligned} \tag{2.1}$$

where $\forall k \; g_k(x) \, \mathbb{R}^n \mapsto \mathbb{R}$ is a differentiable function. Evidently, in (2.1) – with the exception of singular cases– the constraint equations determine an $n-K$ dimensional hypersurface embedded in $\mathbb{R}^n$ over which the actual minimum has to be found.

The basic idea of complexity reduction is very simple: for computing the reduced gradient it is not necessary to invest effort into computing the Lagrange multipliers defined in (2.2).

$$\forall k: \;\; -\left(\frac{\partial g_k}{\partial x}\right)^T \frac{\partial f}{\partial x} + \sum_{\ell} \left(\frac{\partial g_k}{\partial x}\right)^T \frac{\partial g_\ell}{\partial x} \lambda_\ell = 0 \;\; . \tag{2.2}$$

Instead of that it is enough to put the appropriate gradients into the columns of a matrix as

$$\left[\ \frac{\partial g_1}{\partial x}\ \middle|\ \dots\ \middle|\ \frac{\partial g_K}{\partial x}\ \middle|\ -\frac{\partial f}{\partial x}\ \right] \tag{2.3}$$

and apply to this matrix the Gram-Schmidt algorithm [74, 75] that originally was invented by Laplace [76]. It is based on the observation that if we have two linearly independent vectors a and b then with some real factor $\nu \in \mathbb{R}$ the vector $\tilde{b} := b - \nu a$ can be made orthogonal to a. Really, orthogonality means that $a^T b - \nu a^T a = 0$, i.e., $\nu = a^T b / a^T a$, and $b = \tilde{b} + \nu a$. Accordingly, in the first step the $2^{\text{nd}}, \dots, K+1^{\text{th}}$ columns of the matrix in (2.3) must be made orthogonal to the first column in this manner. In the next step the $3^{\text{rd}}, \dots, K+1^{\text{th}}$ columns of the so obtained matrix must be orthogonal to its 2^{nd} column, and so on. Finally, in the last column a vector appears that is orthogonal to each previous column of the final matrix. Since the first K columns of the original matrix in (2.3) can be built up by linearly combining the first K columns of the final matrix, each of them will be orthogonal to the first K columns of (2.3), that is, in the last column of the final matrix the reduced gradient appears, and for this it is not necessary to build up the Lagrange multipliers defined in (2.2). Furthermore, if the norm of a column becomes very small, it simply can be skipped in the process to avoid the appearance of very big ν factors.

Further simplification can be done if we take it into account that $\forall i\, g_i(x) = 0$ if and only if $G(x) = 0$. Consequently, it is enough to make a single step of orthogonalization for the two column matrix $\left[\frac{\partial G}{\partial x}\middle|\frac{-\partial f}{\partial x}\right]$.

Finally, instead of the original form of $G(x)$ its modified version $G(x) := \sum_\ell |g_\ell(x)|^\beta$ can be considered for $1 < \beta < 2$ based on the observation that for the function $|x|$ the Newton–Raphson algorithm would be very fast, however, around the $x = 0$ it should have instabilities. This instability would be evaded for a β that is a little bit greater than 1. On the same time it is not trivial if the calculation of the $|x|^\beta$ powers could result in time-sparing. For this purpose numerical investigations, experiments must be done.

2.1.1 A Simple Numerical Example

Since in [77, 78] the basic idea was the use of non-quadratic cost functions that can penalize the error terms in sophisticated manner, depending on the inner parameters of these cost functions, for numerical investigations the following simple problem was considered in [79]:

$$\begin{aligned}
\text{minimize} \quad & f(x) = A_1(x_{n1} - x_1)^2 + \\
& A_2(x_{n2} - x_2)^4 + A_3(x_{n3} - x_3)^6 \\
\text{s.t.} \quad & g_1(x) = 0, \quad g_2(x) = 0 \\
\text{where} \quad & g_1(x) = x_1^2 + x_2^2 + x_3^2 - R^2, \\
& g_2(x) = (x_1 - 0.5R)^2 + x_2^2 + x_3^2 - R^2 \\
& x_{1n} = 0.2R \ , \quad x_{2n} = 0.3R \ , \quad x_{3n} = 0.4R
\end{aligned} \tag{2.4}$$

The optimization was commenced from the known absolute minimum of the cost function. From this point at first the Newton–Raphson algorithm moved the starting point to the constraint surface then the reduced gradient algorithm continued moving over the surface. The parameters of the computations are given in Table 2.1.

Table 2.1: The parameters of the numerical calculations

Parameter	**Value**
Constraint radius R	5.0
Step length for gradient estimation δx	10^{-3}
Error limit ε	10^{-2}
Newton–Raphson speed parameter a_1	0.5
Reduced Gradient speed parameter a_2	10^{-3}
Cost function parameter A_1	1.0
Cost function parameter A_2	2.0
Cost function parameter A_3	3.0
Cost function parameter β	1.1

In the first step the results obtained for the sequential gradient reduction with $\partial g_1/\partial x$ and $\partial g_2/\partial x$ are compared with that obtained by a single reduction with $\partial G/\partial x$ for $G = g_1(x)^2 + g_2(x)^2$.

According to the expectations both algorithms found the same local optimum with the same final cost function with quite comparable number of numerical steps. In both cases the same Newton–Raphson algorithm was used during 30 steps while the cost value f increased from its absolute minimum 0 to about the value of 140, and the constraint variable G well approximated 0. The appropriate cost functions are described in Fig. 2.1.

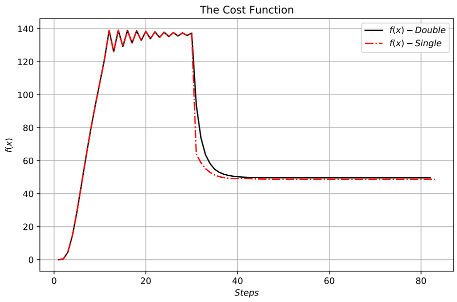

Figure 2.1: Variation of the cost function for the double steps and the single step gradient reduction for $G = g_1(x)^2 + g_2(x)^2$

Following that a Reduced Gradient Algorithm kept the $G = 0$ or the $g_1(x) = 0$, and $g_2(x) = 0$ values, pushed the norm of the reduced gradient to zero and reduced the cost value to about 50 (Fig. 2.2).

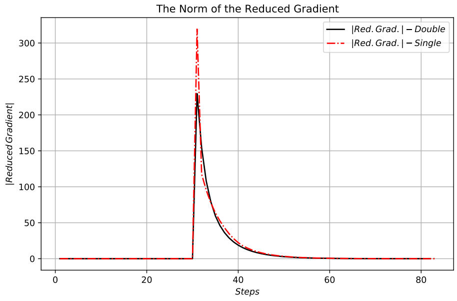

Figure 2.2: Variation of the norm of the reduced gradient (it is relevant only in the second phase of the algorithm) for the double steps and the single step gradient reduction for $G = g_1(x)^2 + g_2(x)^2$

The significant differences are present in the different time-need of the calculations as is given in Fig. 2.3. The main computation need in the Newton–Raphson phase consists of two essential steps: the estimation of the gradient and making the corrective steps. These steps were made by the code provided in the Appendix 7.
While Newton-Raphson phase had the same time-need $4\,[\mu s]$ in both cases, the time-need of the Reduced Gradient phase was reduced from about 75–$100\,[\mu s]$ to 18–$32[\mu s]$ in the case of the single step reduction. Taking into account that in this simple example we had only 2 Lagrange multipliers, generally it can be expected that by reducing the number of these multipliers in more complicated cases more time can be spared by this approach.

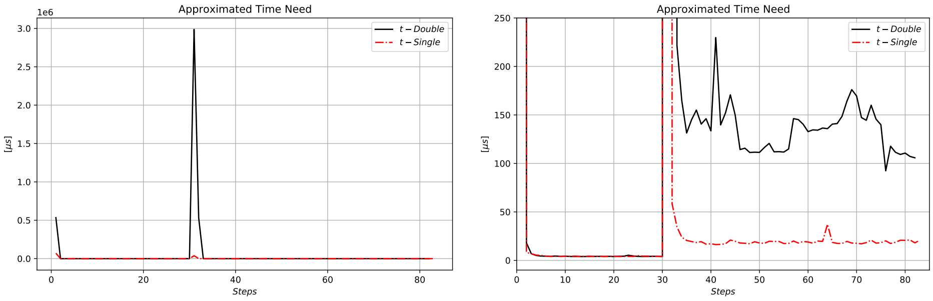

Figure 2.3: The time-need (LHS) and its zoomed (RHS) for the double steps and the single step gradient reduction for $G = g_1(x)^2 + g_2(x)^2$

It is interesting to see how many computational effort is needed in the reduced gradient phase of the algorithm to pull back the actual point onto the constraint surface. Fig. 2.4. reveals that these corrections are significant only at the beginning of the algorithm at which the reduced gradient has relatively great norm.

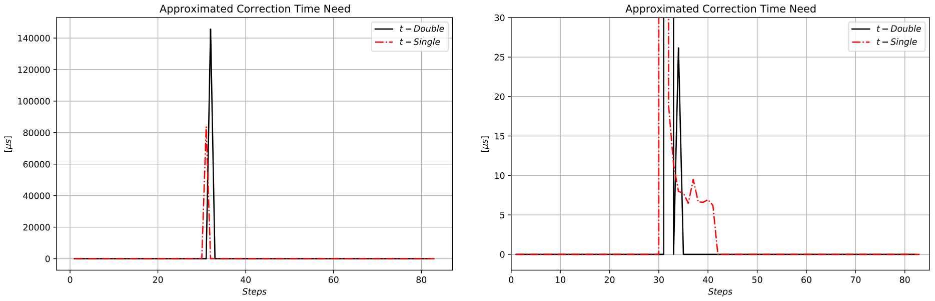

Figure 2.4: The time need (LHS) and its zoomed (RHS) of the corrections needed for pulling back the actual point onto the constraint surface in the reduced gradient phase of the algorithm

In the second step the use of the single constraint term in a single step gradient reduction $G(x) := \sum_\ell |g_\ell(x)|^\beta$ was investigated. It can be stated that the results given in Figs. 2.5, 2.6 are comparable with that of the single step reduction algorithm using $G(x) = g_1(x)^2 + g_2(x)^2$.

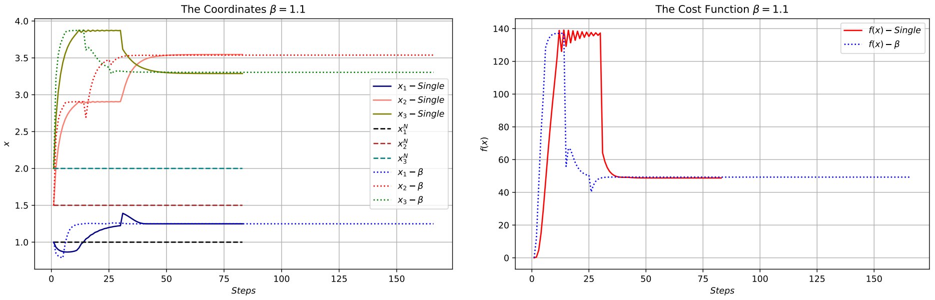

Figure 2.5: Variation of the x coordinates and the value of the cost function for a single step reduction for $G_{Single}(x) = g_1(x)^2 + g_2(x)^2$ against $G_\beta(x) = |g_1(x)|^\beta + |g_2(x)|^\beta$

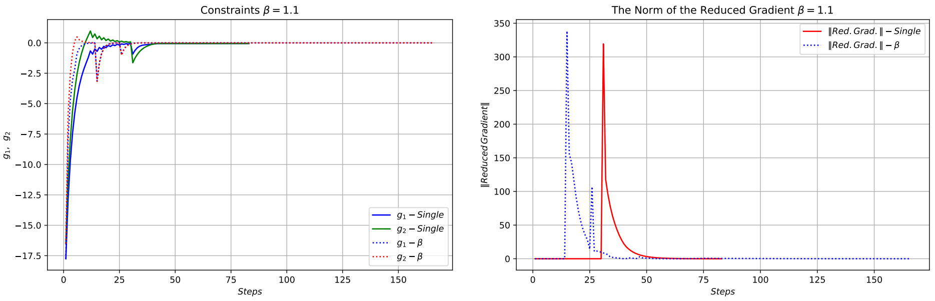

Figure 2.6: Variation of the individual constraint functions and the norm of the reduced gradient function for a single step reduction for $G_{Single}(x) = g_1(x)^2 + g_2(x)^2$ against $G_\beta(x) = |g_1(x)|^\beta + |g_2(x)|^\beta$

However, Fig. 2.8 testifies that considerable increase happened in the computational time.

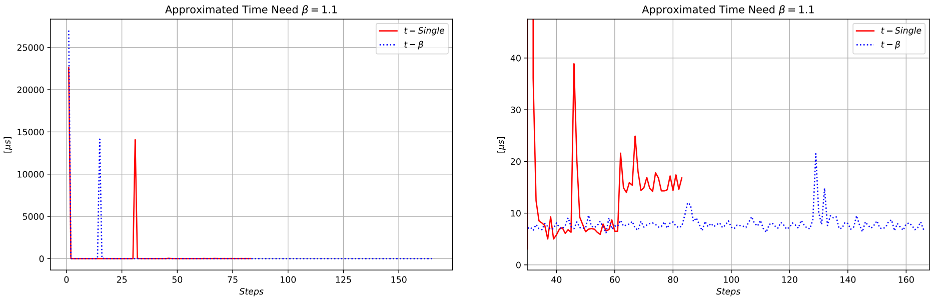

Figure 2.7: The time need of the Newton-Raphson phase and the reduced gradient phase for a single step reduction for $G_{Single}(x) = g_1(x)^2 + g_2(x)^2$ against $G_\beta(x) = |g_1(x)|^\beta + |g_2(x)|^\beta$

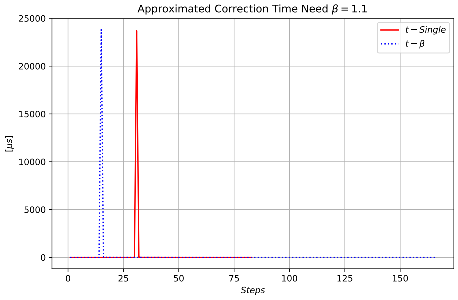

Figure 2.8: The time need of the corrections needed for pulling back the actual point onto the constraint surface in the reduced gradient phase of the algorithm for a single step reduction for $G_{Single}(x) = g_1(x)^2 + g_2(x)^2$ against $G_\beta(x) = |g_1(x)|^\beta + |g_2(x)|^\beta$

In the sequel this simple idea was applied in solving a more complicated task, namely providing the solution of a differential inverse kinematic task of a redundant robot arm.

2.2 Solving the Inverse Kinematic Task by Accelerated Reduced Gradient Algorithm

It is well known that normally the inverse kinematic task of redundant robot arms – with the exception of very special constructions – cannot be solved by the use of simple formulae. Various generalized inverses can be used for selecting a particular one of the huge set of ambiguous solutions. While the Singular Value Decomposition (SVD) [80]-based methods, and the Gram-Schmidt algorithm [74, 75]-based ones may result in velocity profiles that cause huge $\ddot{q}$ values that is difficult to track by the robot as a Classical Mechanical system, the Moore-Penrose pseudoinverse [81, 82] can yield quite smooth solution if the actual configuration is not in the vicinity of singularities. Its main idea is the realization of the necessary motion of the end effector by minimized joint coordinate time-derivatives. In this case the more general optimization task given in (2.1) takes the more special form as follows:

$$\begin{aligned} &\text{minimize } C(\dot{q}) := \sum_{\ell=1}^{n} \dot{q}_\ell^2 \\ &\text{under the constraint } \dot{x} = J(q)\dot{q} \ . \end{aligned} \tag{2.5}$$

The Jacobian of the system $J(q)$ depends on the particular structure and the parameters of the robot arm and on the free choice how the **home position** – outlined in Fig. 2.9– is defined for the given arm. The very simple case study considered in [57] revealed that instead individually setting constraint equations to the constraints $\{g^{(i)}(\dot{q}) = 0\}$ and using only a single constraint $G(\dot{q}) := \sum_\ell g^{(\ell)}(\dot{q})^2 = 0$ quite considerable computational need can be spared. Furthermore, since this approach does not wish to produce some generalized inverse, it arbitrarily can be modified by using various cost functions while the same constraints are considered as in the case of the Moore-Penrose pseudoinverse. In the present section this method is applied for solving the inverse kinematic task for an open kinematic chain structure outlined in Fig. 2.9.

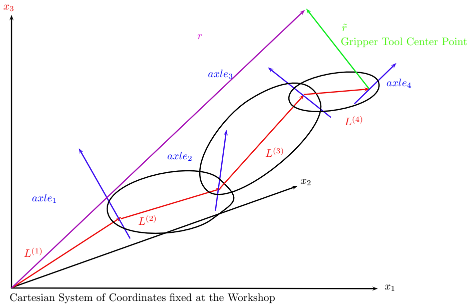

Figure 2.9: Symbolic description of the Home Position of an open kinematic chain (courtesy by J.K. Tar from a not published and not reviewed lecture notes)

2.2.1 Definition of the Kinematic Parameters and the Differential Inverse Kinematic Task

The Inverse Kinematic Task of a robot can be described by the desired position of the *Tool Center Point (TCP)* r with respect to the Cartesian Workshop Coordinates of Reference, and the desired pose O of the workpiece or the tool that can be achieved by various joint coordinates $\xi_1, \xi_2, \cdots, \xi_n$ (it is an ambiguous solution). The velocity of the TCP is given by

(2.6)

$$\begin{gathered}\begin{bmatrix}\dot{r}(t)\\ 0\end{bmatrix}=\left(\dot{\xi}_1\frac{\mathrm{d}H^{(1)}}{\mathrm{d}\xi_1}H^{(1)^{-1}}+\right.\\ \dot{\xi}_2 H^{(1)}\frac{\mathrm{d}H^{(2)}}{\mathrm{d}\xi_2}H^{(2)^{-1}}H^{(1)^{-1}}+\ldots+\\ +\dot{\xi}_n H^{(1)}H^{(2)}\cdots\\ \cdots H^{(n-1)}\frac{\mathrm{d}H^{(n)}}{\mathrm{d}\xi_n}H^{(n)^{-1}}H^{(n-1)^{-1}}\\ \left.\cdots H^{(2)^{-1}}H^{(1)^{-1}}\right)\begin{bmatrix}r(t)\\ 1\end{bmatrix}\ ,\end{gathered}\tag{2.6}$$

in which the $H^{(i)}$ homogeneous matrices are defined as follows:

$$H^{(i)}\left(\xi_i,e^{(i)},L^{(i)}\right)\equiv\begin{bmatrix}O\left(\xi_i,e^{(i)}\right) & L^{(i)}\\ 0^T & 1\end{bmatrix}\tag{2.7}$$

for rotational joint in which the components of the unit vector $e^{(i)}$ are distributed in the generator of the rotation $\mathcal{G}=\begin{bmatrix}0 & -e_3 & e_2\\ e_3 & 0 & -e_1\\ e_2 & e_1 & 0\end{bmatrix}$, and the appropriate rotational matrix is computed by the Rodrigues formula [83] as

$$O^{(i)}=I+\sin\xi_i\mathcal{G}+(1-\cos\xi_i)\mathcal{G}^2\ .\tag{2.8}$$

For shifts

$$H^{(i)}\left(\xi_i,e^{(i)},L^{(i)}\right)=\begin{bmatrix}I & \xi_i e^{(i)}+L^{(i)}\\ 0^T & 1\end{bmatrix}\tag{2.9}$$

is obtained. In (2.6) the upper left block of size 3×3 determines the skew symmetric rotational velocity of the object gripped by the robot. In this equation the element of a linear space, the tangent space of the group of the homogeneous matrices at the identity matrix, that describes the motion of the workpiece $G(t)$, is expressed as a linear combination of the vectors of this linear space $\left\{G^{(i)}(\xi(t))\right\}$ expressed by the joint coordinate velocities $\dot{\xi}_i$ in (2.10).

$$\sum_{i=0}^{n}\dot{\xi}_i G^{(i)}(\xi(t))=G(t)\ .\tag{2.10}$$

For getting rid of unnecessary redundancies the independent components can be placed in the Jacobian of the problem (the upper left block of $G^{(i)}$ is always skew-symmetric, and the 4th row is always zero), the independent elements of $G^{(i)}(\xi(t))$ and that of $G(t)$ can be placed into a column containing 6 rows as:

$$
\begin{aligned}
&G^{(i)} \Leftrightarrow \mathrm{I\!R}^6 \ni J^{(i)} = \begin{bmatrix} G^{(i)}_{12} \\ G^{(i)}_{13} \\ G^{(i)}_{23} \\ G^{(i)}_{14} \\ G^{(i)}_{24} \\ G^{(i)}_{34} \end{bmatrix} \quad , \\
&G \Leftrightarrow \mathrm{I\!R}^6 \ni \dot{x} = \begin{bmatrix} G_{12} \\ G_{13} \\ G_{23} \\ G_{14} \\ G_{24} \\ G_{34} \end{bmatrix} \quad , \mathrm{I\!R}^n \ni \dot{\xi} = \begin{bmatrix} \dot{\xi}_1 \\ \vdots \\ \dot{\xi}_n \end{bmatrix} \quad ,
\end{aligned}
\tag{2.11}
$$

Following that some solution has to be found for the set of redundant linear equations given in (2.11) that also can be written as $\dot{x} = J(\xi)\dot{\xi}$. In the sequel the method suggested in [57] will be applied with the application of a more complex cost function than that of the Moore-Penrose pseudoinverse.

2.2.2 Setting the Cost Function and the Simulation Parameters

Because no any generalized inverse was needed, the following complex cost function form was used:

$$
\begin{aligned}
&C(\xi, \dot{\xi}) = \sum_i C_i(\xi_i, \dot{\xi}_i) \quad , \\
&C_i(\xi_i, \dot{\xi}_i) = \\
&= \begin{cases}
C_i \left| \frac{\dot{\xi}_i}{\dot{\Delta}} \right|^{\dot{P}_i} \left(1 + \frac{\xi_i - \Delta\xi}{\Delta_i} \right)^{P_i} \text{ if } \dot{\xi}_i > 0 \text{ and } \xi_i > \Delta_i \quad , \\
C_i \left| \frac{\dot{\xi}_i}{\dot{\Delta}} \right|^{\dot{P}_i} \text{ if } \dot{\xi}_i < 0 \text{ and } \xi_i > \Delta_i \quad , \\
C_i \left| \frac{\dot{\xi}_i}{\dot{\Delta}} \right|^{\dot{P}_i} \left(1 + \frac{-\xi_i + \Delta\xi}{\Delta_i} \right)^{P_i} \text{ if } \dot{\xi}_i < 0 \text{ and } \xi_i < -\Delta_i \quad , \\
C_i \left| \frac{\dot{\xi}_i}{\dot{\Delta}} \right|^{\dot{P}_i} \text{ if } \dot{\xi}_i > 0 \text{ and } \xi_i < -\Delta_i \quad , \\
C_i \left| \frac{\dot{\xi}_i}{\dot{\Delta}} \right|^{\dot{P}_i} \text{ if } -\Delta_i \leq \xi_i \leq \Delta_i \quad ,
\end{cases}
\end{aligned}
$$

that has the following interpretation: within the safe range $[-\Delta_i, \Delta_i]$ $|\dot{\xi}_i|$ is limited in a similar way as in the case of the Moore-Penrose pseudoinverse (only the power term can be different), but any attempt to leave this region is seriously punished while turning back to the safe range remains without extra punishment. The seriousness of the punishment depends on the parameter $\dot{\Delta}_i$. The Newton-Raphson algorithm was simply realized as

$$
\dot{\xi}(i+1) = \dot{\xi}(i) - \alpha_1 \frac{G(\dot{\xi}(i))}{\|\nabla G(\dot{\xi}(i))\|^2} \nabla G(\dot{\xi}(i)) \tag{2.12}
$$

with $\alpha_1 = 10^{-1}$, and it was stopped when a small value of G, i.e., the situation $G(\dot{\xi}) \leq \mu = 5 \times 10^{-4}$ was achieved. In the reduced gradient phase the iteration happened according to the rule

$$\dot{\xi}(i+1) = \dot{\xi}_i(i) - \alpha_2 \frac{\widetilde{\partial C(\xi,\dot{\xi})}}{\partial \dot{\xi}} \tag{2.13}$$

with $\alpha_2 = 10^{-4}$, and it was stopped when the norm of the reduced gradient met the condition $\left\| \frac{\widetilde{\partial C(\xi,\dot{\xi})}}{\partial \dot{\xi}} \right\| \leq \mu = 5 \times 10^{-4}$. For getting rid of infinite runtime both algorithms were stopped if the number of iterations achieved the 500 step, and they were not stopped due to the above conditions. The parameters α_1 and α_2 were set experimentally for the given simulations.

2.2.3 Simulation Results

For testing the suggested method, a higher DOF system was used so that the inverse kinematics required more calculations to show how the suggested method behaves. Therefore a 7-DOF redundant robot arm was used so that the type of the joints were $[r, r, r, s, r, r, r]$ where r is related to the rotational joint, whereas s to the prismatic (shifting) one. The homogeneous coordinates $\tilde{r} = [1, 0, 0, 1]$ corresponded to the vector connecting the arbitrarily selected point at the last axle with the (TCP) in the home position.

The orientation was described by an orthogonal matrix, since the orientation errors generally can be considered as the relevant components of a near identity orthogonal matrix that –according to the Rodrigues formula– in the lowest order in the angle of rotation:

$$\begin{aligned} O^{goal} &\approxeq (I + G\xi) O^{act} \quad , \\ \left[O^{goal} - O^{act} \right] O^{act^{-1}} &\approxeq G\xi \quad . \end{aligned} \tag{2.14}$$

where G is a skew symmetric matrix of which only 3 elements are independent. These independent components describe the orientation error.

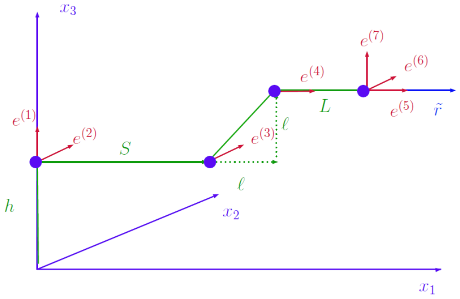

Figure 2.10: The kinematic structure and kinematic parameters of the 7-DOF robot defined in the home position

The unit vectors of the Rotational Axles in the Home Position are given in the columns of the matrix in (2.15)

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} , \tag{2.15}$$

and the shift parameters were given in (2.16)

$$R = \begin{pmatrix} 0 & 0 & S & l & L & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h & 0 & 0 & l & 0 & 0 & 0 \end{pmatrix} . \tag{2.16}$$

Table 2.2: The parameters of the simulation

The Reduced Gradient Method’s parameters	**Value**
Error parameter μ	5×10^{-4}
Newton-Raphson speed parameter α_1	10^{-1}
LGR speed parameter α_2	10^{-4}
Displacement for gradient est. δx [rad] or [m]	10^{-3}
Cost Function Parameters	**Value**
Safe range parameter Δ_i [rad] or [m] $\forall i$	0.8
Velocity limit parameter $\dot{\Delta}_i$ [rad s^{-1}] or [m s^{-1}] $\forall i$	5×10^{-2}
Power parameter for distance P_i $\forall i$	1.5
Power parameter for velocity $\dot{P}_i$ $\forall i$	4
Cost parameters $[C_1, C_2, C_3, C_4, C_5, C_6, C_7]$	$[1, 6, 1, 8, 2, 8, 1]$
Kinematic Robot Parameters	**Value**
The height of the 1^{st} horizontal axle h [m]	1.0
For the 1^{st} long arm S [m]	2
For the 2^{nd} long arm l [m]	1
For the prismatic arm L [m]	0.5

In the simulation, two types of trajectories were investigated. In the first step the Simple Trajectory was considered (Figs. 2.11 – 2.13) that was generated by the formula $Ampl * time$ in the space of the joint coordinates. Here, the trajectory tracking worked nicely with very small error in the range of $-0.45[mm]$ in negative direction and $+0.01[mm]$ in positive one with regard to the Cartesian Coordinates.

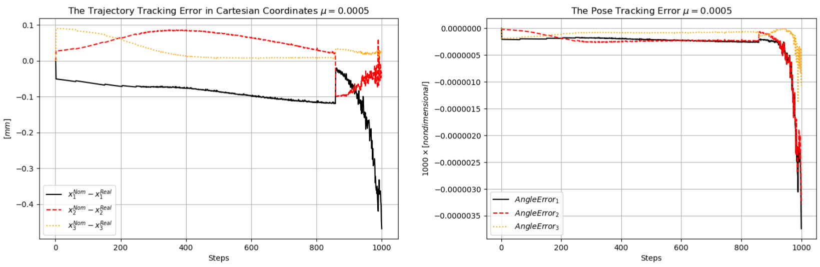

Figure 2.11: The trajectory tracking error (LHS) and the orientation tracking error in Cartesian coordinates (RHS) - simple trajectory

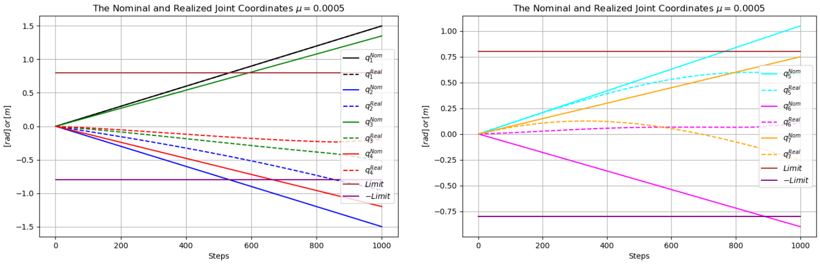

Figure 2.12: The joint coordinates in tracking the simple trajectory

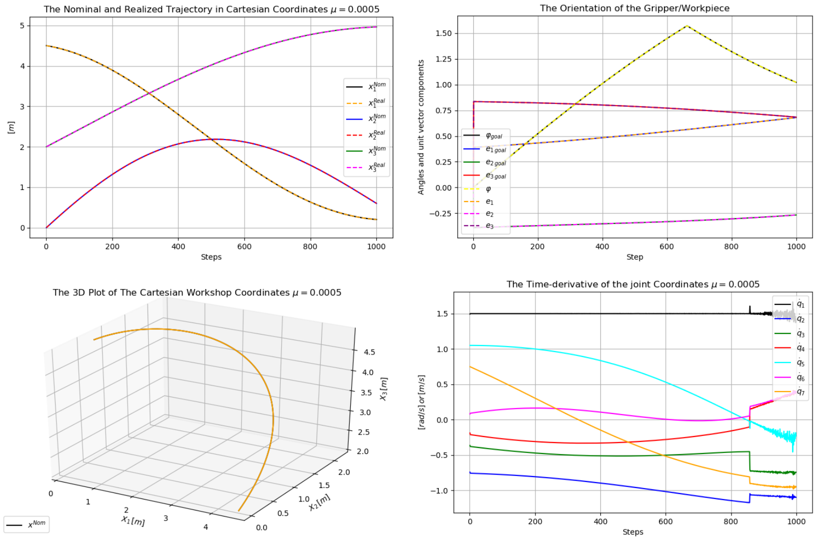

Figure 2.13: At the top: nominal and realized Cartesian coordinates and the orientation of the tool, at the bottom: 3D picture of the Cartesian trajectory, and the joint coordinate velocities - simple trajectory

The other trajectory was the Complex Trajectory that was generated in the space of the joint coordinates by the formula $10 * Ampl.sin.(\Omega * time)$ in which $Ampl$ and Ω are 1×7 matrices: each matrix element contains the parameters of the sinusoidal nominal motion for the appropriate joint coordinate. This complex trajectory sometimes approached kinematic singularities that were observed by increased Cartesian and orientation errors (as it can be seen in Fig. 2.14).

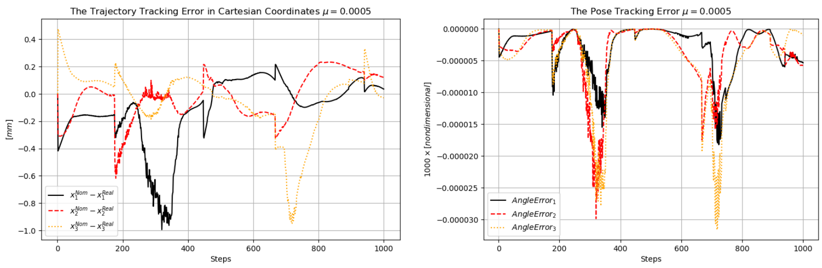

Figure 2.14: The trajectory tracking error (LHS) and the orientation tracking error in Cartesian coordinates (RHS) - complex trajectory

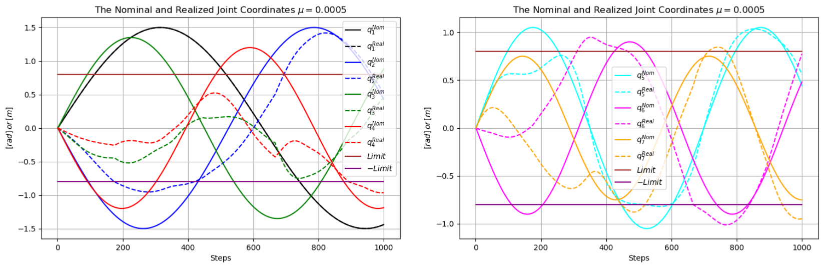

Figure 2.15: The joint coordinates in tracking the complex trajectory

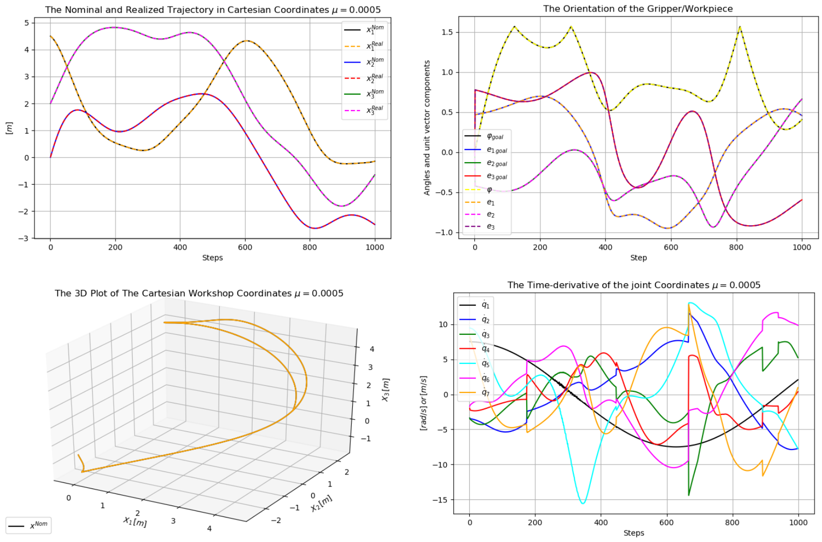

Figure 2.16: At the top: nominal and realized Cartesian coordinates and the orientation of the tool, at the bottom: 3D picture of the Cartesian trajectory, and the joint coordinate velocities - complex trajectory

It can be observed that big jump (which means big punishment from the cost function) can happen when the realized trajectory for any joint hits the lower permitted limits and where the velocity is in the appropriate (positive or negative) direction. For example, the first jump happened with the joint 2 at step 173 which is described in Fig. 2.17. The second one happened when the joint 6 reached the step 665, there, an immediate jump happened so that correcting the direction and avoiding dangerous position as in Fig. 2.18. However, the event in which joint 2 achieved the upper limit about the step number approx. 690 in Fig. 2.15, did not generate such drastic jump in the joint velocities. This fact can be explained by the interaction between the motion of the various axles i.e. that the inverse kinematic task has to be solved even at high costs. This behavior originates from the fact that in the task defined in (2.2.2) the constraint equations are precisely realized even near the singularities where large joint velocity components have to occur. In these cases the costs simply are increased and the cost function cannot efficiently act as a penalizing factor.

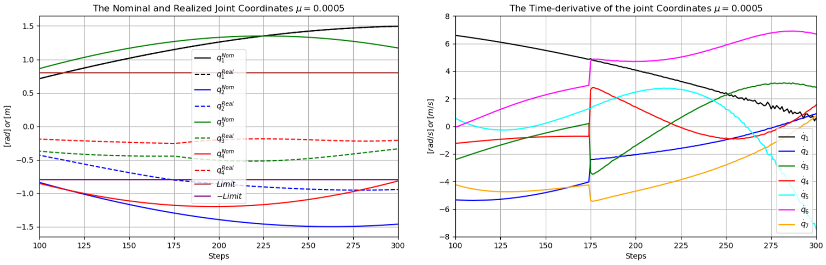

Figure 2.17: The jump in Joint Coordinates Joint velocities (LHS) related to the Joint 2 (RHS)

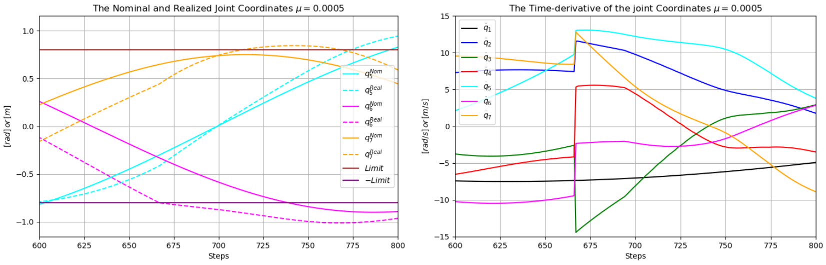

Figure 2.18: The biggest jump in Joint Coordinates Joint velocities (LHS) related to the Joint 6 (RHS)

2.3 Thesis Statement I

I have recognized that in contrast to the traditional approach of optimal controllers, in which the optimization of a cost function happens via individually dealing with each dynamic model term as a constraint equation, it is possible to construct a single constraint equation that guarantees the fulfillment of each original constraint. In this new approach, similarly to the traditional one, the Newton-Raphson algorithm is used for finding a point on the embedded hypersurface that contains the possible solutions, and Lagrange's Reduced Gradient Algorithm is used for moving along the hypersurface, but in our case only one Lagrange multiplier can be used. By directly using the Gram-Schmidt algorithm for gradient reduction, in this manner considerable decrease in the

computational efforts became possible.

I have observed that in application areas that contain singularities (e.g., in solving inverse kinematic task of redundant robot arm) this approach is sensitive to the presence of kinematic singularities as well as the Moore-Penrose pseudoinverse-based solutions that need the application of complementary tricks for dealing with near singularity solutions. I have recognized that the common reason of this sensitivity is that the constraints are exactly taken into account in both cases.

Related own publications: **[A. 1] and [A. 2]**

Chapter 3

Improvement of the Fixed Point Iteration-based Adaptive Receding Horizon Controller

In this chapter, I have investigated the applicability and limitations of using (ARHC) in various engineering applications. Because the Fixed Point Iteration-based Adaptive Control method is considered in the Thesis, it is expedient to briefly summarize its essence in the sequel.

3.1 The Fixed Point Iteration-based Adaptive Control

The schematic structure of this controller is described in Fig. 3.1.

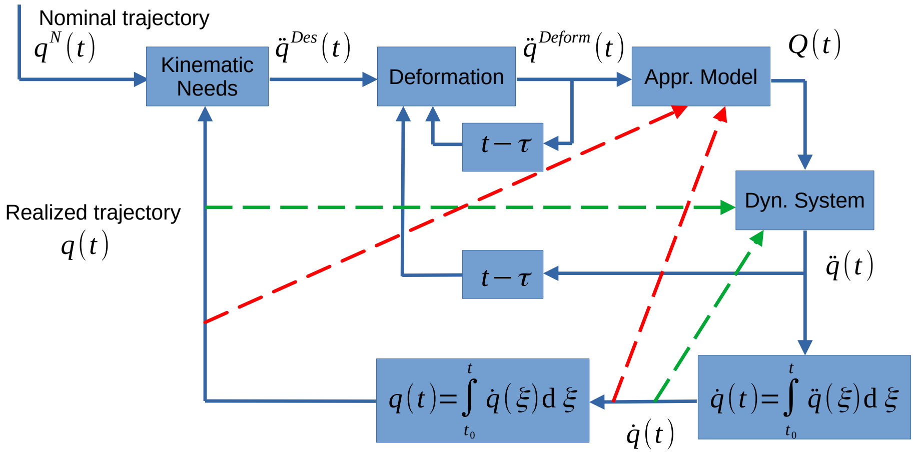

Figure 3.1: The schematic structure of the *Fixed Point Iteration-based Adaptive Controller* for a second order dynamical system (after [84]); the delay time τ normally corresponds to the cycle time of the digital controller.

The boxes of this scheme can be filled in with various contents that may differ from each other in little details. The main issues are detailed in the sequel.

3.1.1 The Response Function

This method is the modification of the Computed Torque Control [11] that directly uses the inverse dynamic model of the controlled system (in the present controller it is in the block *Appr. Model*) for the calculation of the necessary control forces. (Since normally only its approximate version is known, it is referred to as approximate model.) The box *Dyn. System* symbolizes the exact model of the controlled system. These two models realize a composite function

$$\ddot{q} = \mathfrak{E}\left(q, \dot{q}, \mathfrak{A}\left(q, \dot{q}, \ddot{q}^{Deform}\right)\right) \approxeq f(\ddot{q}^{Deform}) \tag{3.1}$$

where $\mathfrak{A}$ denotes the approximate inverse model, and $\mathfrak{E}$ stands for the exact dynamic model. As the dashed lines in the figure indicate it, both depend on the actual state of the 2nd order system. However, since $q(t)$ and $\dot{q}(t)$ only slowly vary in time, while $\ddot{q}(t)$ can be abruptly changed, (3.1) is a good approximation. **The function $f\left(\ddot{q}^{Deform}\right)$ is referred to as the controlled system's response function.**

3.1.2 Kinematic Requirements

The block*Kinematic Needs* can contain various functions of $q^N(t)$ and $q(t)$ the realization of which makes the trajectory tracking error converge to zero if the $\ddot{q}^{Des}$ value recommended by t is realized. Normally it is built up of the components given in (3.2).

$$e(t) := q^N(t) - q(t) \text{ tracking error} \tag{3.2a}$$

$$\dot{e}(t) = \dot{q}^N(t) - \dot{q}(t) \text{ derivative tracking error} \tag{3.2b}$$

$$e_{Int}(t) = \int_{t_0}^{t} e(\xi)\mathrm{d}\xi \text{ integrated tracking error } . \tag{3.2c}$$

For instance, (3.3) corresponds to a PID-type solution, while (3.4) means a PD-type error feedback for a constant, positive Λ.

$$\begin{gathered}\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^3 e_{Int}(t) \equiv 0 \Rightarrow \\ \ddot{q}^{Des}(t) = \ddot{q}^N(t) + \Lambda^3 e_{int}(t) + 3\Lambda^2 e(t) + 3\Lambda\dot{e}(t) \ .\end{gathered} \tag{3.3}$$

$$\begin{gathered}\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^2 e(t) \equiv 0 \Rightarrow \\ \ddot{q}^{Des}(t) = \ddot{q}^N(t) + \Lambda^2 e(t) + 2\Lambda\dot{e}(t) \ .\end{gathered} \tag{3.4}$$

Regarding the proof, it is easy to see that the *general solution of the time-invariant, linear, homogeneous set of equations*

$$\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^m g(t) \equiv 0 \tag{3.5}$$

take the form

$$g(t)=\sum_{\ell=0}^{m-1} c_\ell (t-t_0)^\ell \exp\left(-\Lambda(t-t_0)\right) \text{ for } t>t_0 \ . \tag{3.6}$$

where the $\{c_\ell\}$ coefficients are determined by the initial conditions. With other words, the linear space of the solutions of (3.6) is spanned by the basis functions, and each of these basis functions converges to zero as $t\to\infty$. The polynomial functions in (3.6) may cause some fluctuation of the errors in the PID-type control, and a less complex initial variation in the PD type one. I note that it is not compulsory to deduce the *proportional*, the *derivative* and the *integrated* error feedback gains from a single parameter Λ. To evade the transient fluctuations often *fractional order derivatives* are fed back as e.g., in [85–88]. More general info on fractional order derivatives can be found e.g., in [89, 90].

3.1.3 Convergence Issues

Since the available inverse model usually is not precise, function $\mathfrak{A}$ is not exactly the inverse of $\mathfrak{C}$. For achieving the desired kinematic design as the output of the response function as $\ddot{q}^{Des}=f(\ddot{q}_\star)$, in the input argument a deformed value $\ddot{q}_\star$ must be written. To find the appropriate extent of deformation an iterative process can be invented in the form as follows for an $\alpha\in\mathbb{R}$ ($\ddot{q}\equiv x$):

$$x_{n+1}^{Def}=x_n^{Def}+\alpha\left(x_n^{Des}-f\left(x_n^{Def}\right)\right) \ . \tag{3.7}$$

In the vicinity of the appropriate deformed value the following approximations can be done:

$$f\left(x_n^{Def}\right)=f\left(x_\star+x_n^{Def}-x_\star\right)\approxeq f(x_\star)+\frac{\partial f}{\partial x}\left(x_n^{Def}-x_\star\right) \ . \tag{3.8}$$

The iteration in (3.7) can be so realized that during one step of digital controller only one iterative step can be realized. However, if x_n^{Des} varies only slowly, this iteration can be convergent. Approximately, to a constant x^{Des} a constant $x_\star$ belongs. Since $f(x_\star)=x^{Des}$ it can be written that:

$$x_{n+1}^{Def}-x_\star\approxeq\left[I-\alpha\left.\frac{\partial f}{\partial x}\right|_{x_\star}\right]\left(x_n^{Def}-x_\star\right) \ , \tag{3.9}$$

that can be convergent if $\alpha>0$ is small, and the function $f(x)$ is *differentially approximately direction keeping*. That means that for a small δx displacement it holds that:

$$\delta f^T\delta x=[f(x+\delta x)-f(x)]^T\delta x>0 \ . \tag{3.10}$$

For an arbitrary non-singular real quadratic matrix M and a real column of appropriate size w it can be written that

$$\begin{gathered}\|[I-\alpha M]w\|^2=w^T\left[I-\alpha\left(M+M^T\right)+\alpha^2M^TM\right]w= \\ \|w\|^2-\alpha w^T\left(M+M^T\right)w+\alpha^2w^TM^TMw \ ,\end{gathered} \tag{3.11}$$

that is the Frobenius norm of the transformed column can decrease if

$$\|w\|^2-\alpha w^T\left(M+M^T\right)w+\alpha^2w^TM^TMw<\|w\|^2 \ , \tag{3.12}$$

that for $\alpha > 0$ leads to

$$\begin{aligned} 0 &< \alpha w^T M^T M w < w^T (M + M^T) w \\ 0 &< \alpha < \frac{w^T (M + M^T) w}{w^T M^T M w} \end{aligned} \tag{3.13}$$

since $0 < w^T M^T M w$ can be assumed.[1]

In the control of mechanical devices, if the inverse of the inertial matrix is well approximated, this direction keeping property can be well assumed in the practice for the response function (i.e.,the product of the exact inertia matrix and the inverse of its approximation must be in the vicinity of the identity matrix.)

3.1.4 On the Content of Block Deformation

In the above considerations in the practice it is difficult to know which values of α are small enough for guaranteeing the fact of convergence, and large enough for guaranteeing fast enough convergence. The simple parameter α in (3.7) was only the simplest version of transforming the control task into finding the fixed point of an iteration. For this purpose various functions were suggested for use in this block. In the first relevant publication in [84] the *Robust Fixed Point Transformation* a function containing three parameters was recommended for single variable functions as in (3.14).

$$\begin{aligned} &\ddot{q}^{Def}(t_{n+1}) = \left(K_c + \ddot{q}^{Def}(t_n)\right) \times \\ &\left[1 + B_c \sigma\left(A_c\left(f\left(\ddot{q}^{Def}(t_n)\right) - \ddot{q}^{Des}(t_{n+1})\right)\right)\right] - K_c \end{aligned} \tag{3.14}$$

where $-1 < \sigma(x) < 1$ is a rigorously monotonic increasing (sigmoid) function with $\sigma(0) = 0$, and $\sigma'(0) = 1$, as e.g., $\tanh(x)$. Evidently, if $f\left(\ddot{q}^{Def}(t_n)\right) - \ddot{q}^{Des}(t_{n+1}) = 0$ then the iteration arrives at $\ddot{q}^{Def}(t_{n+1}) = \ddot{q}^{Def}(t_n)$, i.e., the solution is the fixed point of this parametric function. According to Banach's fixed point theorem, if we have a contractive map in linear, normed, complete metric space (Banach space) as $F : \mathscr{B} \mapsto \mathscr{B}$, the iterative sequence generated by this function as $\{\ldots, x_{n+1} = F(x_n), \ldots\}$ converges to the unique fixed point of this function $x_\star = F(x_\star)$. Therefore, the three parameters A_c, K_c, and B_c must be so chosen that the resulting mapping must be contractive. Contractivity means that there exists a $0 \leq L < 1$ limit value so that $\forall x, y \in \mathscr{B}$ $\|F(y) - F(x)\| \leq L\|y - x\|$.

For setting the parameters consider the value of the following derivative in (3.14):

$$\frac{d\ddot{q}^{Def}(t_{n+1})}{d\ddot{q}^{Def}(t_n)} = 1[\ldots] + \left(K_c + \ddot{q}^{Def}(t_n)\right) B_c A_c \sigma'(\ldots) \frac{df}{d\ddot{q}^{Def}(t_n)} \tag{3.15}$$

[1] In quantum mechanical applications it is often utilized that the eigenvalues of Hermitian –generally complex– matrices are real numbers. In control technology this assumption is rarely utilized. Really, by definition a matrix M is Hermitian if $M^{T*} = M$. In this case for the eigenvalue λ and –generally complex– eigenvector x it can be written that $Mx = \lambda x$, and $x^{T*} M x = \lambda x^{T*} x$. This equation is equivalent with $x^{T*} M^{T*} = \lambda^* x^{T*}$, that, because M is Hermitian, leads to $x^{T*} M = \lambda^* x^{T*}$, and $x^{T*} M x = \lambda^* x^{T*} x$. These together lead to $\lambda^* x^{T*} x = \lambda x^{T*} x$ where $x^{T*} x > 0$, that is λ must be real. The real Hermitian matrices are the symmetric ones. Since each real matrix can be decomposed into symmetric and skew symmetric parts as $M = \frac{1}{2}\left(M + M^T\right) + \frac{1}{2}\left(M - M^T\right)$, in the direction keeping property only the symmetric part of the matrix plays role, since due to symmetry reason $w^T\left(M - M^T\right) w = 0$.

in which in the fixed point$\left[\ldots\right] = 1$, and $\sigma'(0) = 1$, therefore it takes the value

$$\frac{\mathrm{d}\ddot{q}^{Def}(t_{n+1})}{\mathrm{d}\ddot{q}^{Def}(t_n)} = 1 + \left(K_c + \ddot{q}^{Def}(t_n)\right) A_c B_c \sigma'(0) \frac{\mathrm{d}f}{\mathrm{d}\ddot{q}^{Def}(t_n)} \quad . \tag{3.16}$$

Let $B_c = \pm 1$, $K_c \gg \left|\ddot{q}^{Def}(t)\right|$. In this case (3.16) can be approximated as

$$\frac{\mathrm{d}\ddot{q}^{Def}(t_{n+1})}{\mathrm{d}\ddot{q}^{Def}(t_n)} \approxeq 1 + K_c A_c B_c \sigma'(0) \frac{\mathrm{d}f}{\mathrm{d}\ddot{q}^{Def}(t_n)} \quad . \tag{3.17}$$

If the sign of $\frac{\mathrm{d}f}{\mathrm{d}\ddot{q}^{Def}(t_n)}$ is well defined in a task, a small A_c value can make (3.17) equal to $1-\varepsilon \in (-1,1)$ that guarantees convergence, since for a single variable, differentiable real function the integral estimation can be used as

$$\begin{aligned} f(y) - f(x) &= \int_x^y f'(\xi)\mathrm{d}\xi \\ |f(y) - f(x)| &\leq \int_x^y \left|f'(\xi)\right| \mathrm{d}\xi \quad , \end{aligned} \tag{3.18}$$

therefore if $\exists 0 \leq L < 1$ so that $|f'(x)| \leq L$, this single variable function is contractive.

For multiple variable functions Dineva suggested appropriate parametric fixed point transformation functions in [91]. It is based on the following simple construction: let $g(x) : \mathrm{I\!R} \mapsto \mathrm{I\!R}$ be a bounded, strictly monotonic increasing function, and $g^{-1}(x)$ its inverse. Let $F(x) : \mathrm{I\!R} \mapsto \mathrm{I\!R}$ defined as $F(x) := g^{-1}(g(x) - K) + D$. Consider the sequence generated by $F(x)$ as $\{\ldots, x_{n+1} = F(x_n), \ldots\}$. For the so obtained sequence it is true that

$$g(x_{n+1} - D) = g(x_n) - K \quad . \tag{3.19}$$

For appropriate parameters D and K this sequence evidently converges to the unique fixed point of function $F(x)$ (see Fig. 3.2 that contains other simple recommendation for $F(x)$).

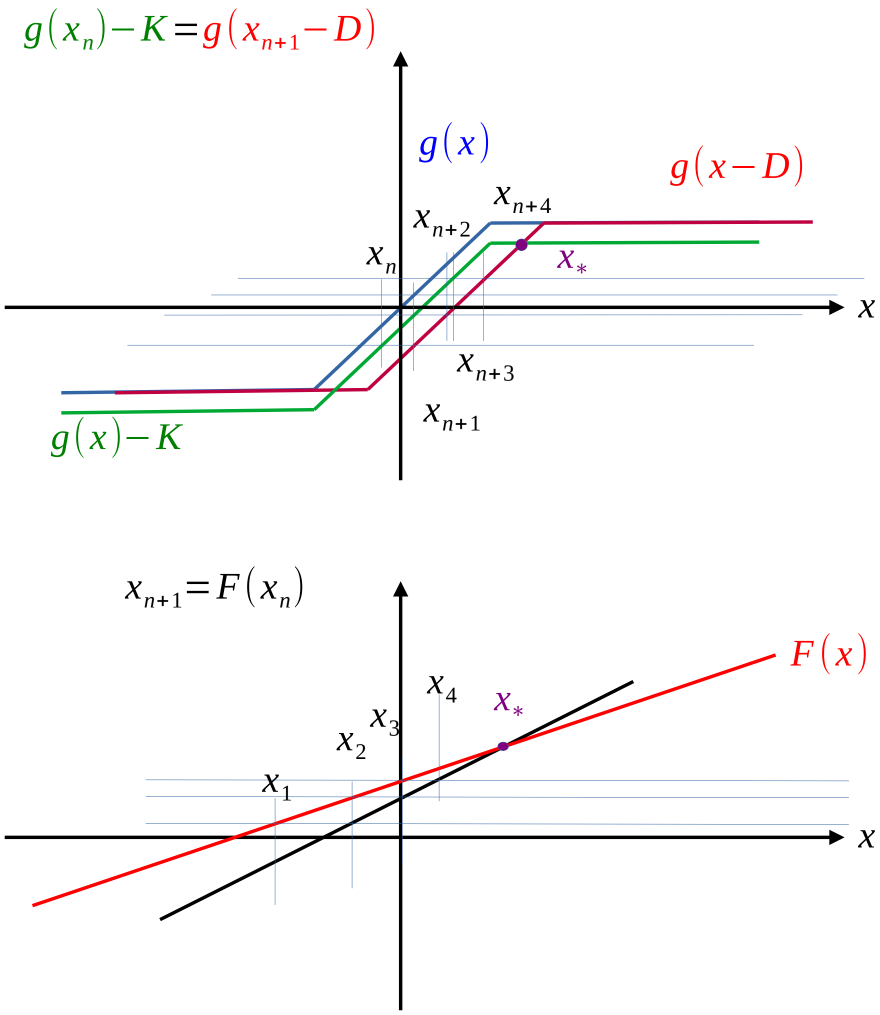

Figure 3.2: Schematic figure explaining the convergence of sequence $\{\dots, x_{n+1} = F(x_n), \dots\}$

Evidently there is a fixed point of $F(x)$ as $x_\star = F(x_\star)$. Dineva suggested the following iteration for the inputs $r \in \mathbb{R}^n$:

$$\begin{aligned} h(i) &:= f(r(i)) - r^{Des} \quad , \\ e(i) &:= \frac{h(i)}{\|h(i)\|} \quad , \\ r(i+1) &:= [F(A\|h(i)\| + x_\star) - x_\star]\, e(i) + r(i) \quad . \end{aligned} \tag{3.20}$$

Evidently, if $h(i) = 0$ then $r(i+1) = r(i)$, that is this function has a fixed point $r_\star \in \mathbb{R}^n$, and only one parameter $A \in \mathbb{R}$ must be set to achieve convergence. For the estimation the requirement for convergence in the vicinity of the fixed point $x_\star \in \mathbb{R}$ $F(x)$ can be approximated by its first order Taylor series expansion, leading to

$$r(i+1) = F'(x_\star) A \|h(i)\| e(i) + r(i) \quad . \tag{3.21}$$

A similar Taylor series approximation can be done for $f(r)$ in the vicinity of $r_\star$ for which it holds that $f(r_\star) = r^{Des}$:

$$f(r) = f(r_\star + r - r_\star) \approxeq f(r_\star) + \left.\frac{\partial f}{\partial r}\right|_{r_\star} [r - r_\star] = r^{Des} + \left.\frac{\partial f}{\partial r}\right|_{r_\star} [r - r_\star] \quad , \tag{3.22}$$

leading to

$$h(i) := f(r(i)) - r^{Des} \approxeq \left.\frac{\partial f}{\partial r}\right|_{r_\star} [r - r_\star] \quad . \tag{3.23}$$

By substituting this into (3.21) it is obtained that

$$r(i+1) \approxeq F'(x_\star) A \|h(i)\| \frac{h(i)}{\|h(i)\|} + r(i) = F'(x_\star) A \left.\frac{\partial f}{\partial r}\right|_{r_\star} [r - r_\star] \quad , \tag{3.24}$$

that is

$$r(i+1) - r_\star \approxeq \left[I + F'(x_\star) A \left.\frac{\partial f}{\partial r}\right|_{r_\star} \right] (r(i) - r_\star) \tag{3.25}$$

It is evident that we have to consider the same contractivity problem as in (3.9).

To tackle the problem of finding the appropriate parameter α in (3.7) in [92] the augmentation of the appropriate arrays was suggested to obtain vectors of identical Frobenius norm, since these vectors can be rotated into each other, and the selected parameter can be a simple interpolation with the angle of rotation as indicated in Fig. 3.3. The physically interpreted projections of the rotated vectors simultaneously suffer rotation and shrink/dilatation, and the appropriate adaptive factor λ_a must be found in the $(0,1)$ interval. For constructing the rotation operators the generalization of the Rodrigues formula [83] was applied.

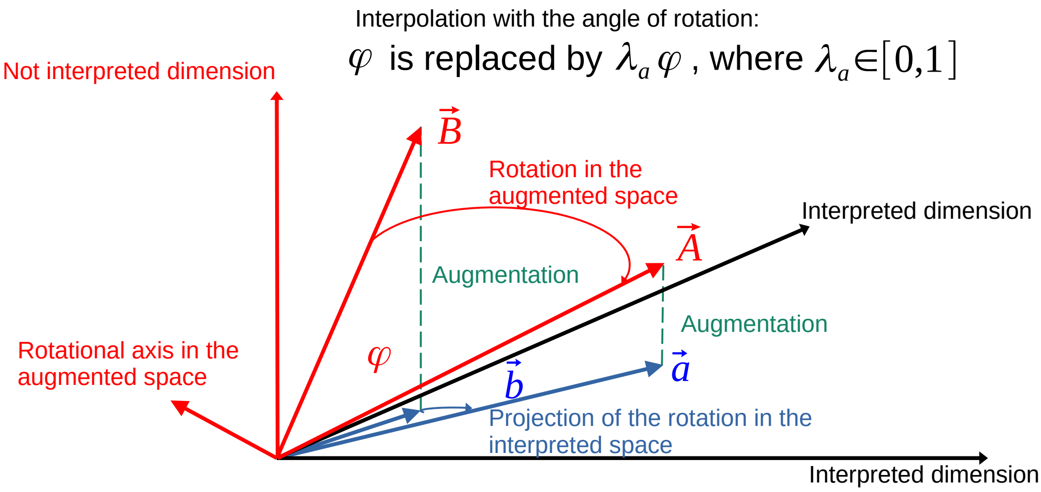
Interpolation with the angle of rotation:
φ is replaced by $\lambda_a \varphi$, where $\lambda_a \in [0,1]$
Not interpreted dimension
$\vec{B}$
Rotation in the augmented space
Interpreted dimension
$\vec{A}$
Augmentation
Augmentation
φ
Rotational axis in the augmented space
$\vec{b}$
$\vec{a}$
Projection of the rotation in the interpreted space
Interpreted dimension

Figure 3.3: The idea of abstract rotation of the augmented vectors (published in [29])

3.2 The Receding Horizon Scheme-based Approach without Gradient Reduction

Generally Lagrange's Reduced Gradient Algorithm [6] is used for finding the optimum under constraints referred to as Nonlinear Programming. Recent investigations [57, 58] revealed that considerable numerical complexity reduction can be achieved by reducing the number of the necessary Lagrangian multipliers. These investigations were discussed in details in Thesis I.

Another formal possibility is completely evading gradient reduction in spite of the fact that the Lagrange multipliers often have important physical interpretation [51], and reveal a strict analogy between the mathematical formalism of the optimal controllers and the Canonical Equations of Motion of Classical Mechanical Systems [5, 52, 53]. From this analogy important mathematical conclusions can be concluded as e.g., the similarity with the flow of incompressible fluids. However, in these considerations the artificial co-state variables (the Lagrange multipliers themselves) take part that are not necessary entities in optimization. In the forthcoming considerations complete evasion of the Lagrange multipliers in optimization task are considered. This idea was also coined and used by Redjimi et al. in [58, 93] with the intent of speeding up computations in state estimation of a particular type 1 diabetes mellitus model, but this possible research direction was not further continued by them. Systematic investigations in utilizing the avoidance of using constraint terms in optimization were made by me as it is summarized in the sequel.

The basic idea of RHC is to calculate a defined *Cost function* for every point over a given finite horizon with respect to the requirements of the described system defined by its dynamic model. The cost function must include the *Control Signal F* and q^o as the *Optimized State Variable* to define the extent of penalization in the cost minimization process. (In the last point of the horizon an extra terminal contribution can be applied that is not used in these considerations.) For a first order dynamical system of equation of motion $\dot{q} = f(q,F)$ the original approach in which the model is taken into account as constraint takes the form of (3.26):

$$\begin{aligned}
&\text{Minimize} \sum_{\ell=1}^{HL} \Phi\left(q^o(\ell), F(\ell)\right) \\
&\text{under the constraints} \, \forall i \in \{1, \dots, HL-1\}: \\
&\frac{q^o(i+1) - q^o(i)}{\Delta t} - f\left(q^o(i), F(i)\right) = 0 \\
&\text{by varying} \, \{F_1, \dots, F_{HL-1}\} \, \text{and} \, \{q_2, \dots, q_{HL}\} \;,
\end{aligned} \tag{3.26}$$

where $\Phi\left(q^o(\ell), F(\ell)\right)$ denotes the cost function in grid point ℓ, Δt is the time-resolution of the grid, and $HL \in \mathbb{N}$ is the horizon length, and $q^o(1)$ is ab ovo given as the *initial state* of the controlled system. Taking into account the dynamic model as constraint

This mathematical complexity evidently can be evaded by incorporating the dynamic model into the function that directly calculates the costs. In this case Evidently, this idea can be extended easily for a second (and higher) order dynamical system in the manner as follows:

$$
\begin{aligned}
&\text{Minimize} \sum_{\ell=3}^{HL} \Phi\left(q^o(\ell), F(\ell)\right) \text{ over the grid points} \\
&\ddot{q}^o(i) = f(q^o(i), \dot{q}^o(i), F(i)) \forall i \in \{1, \dots, HL-2\} \ , \\
&\dot{q}^o(i) = \dot{q}^o(i-1) + \Delta t \ddot{q}^o(i-1) \forall i \in \{2, \dots, HL-1\} \ , \\
&q^o(i) = q^o(i-1) + \Delta t \dot{q}^o(i-1) \forall i \in \{3, \dots, HL\} \ , \\
&\text{by varying } \{F(1), \dots, F(HL-2)\} \ ,
\end{aligned}
\tag{3.27}
$$

in which $q^o(1)$, $q^o(2)$, and $\dot{q}^o(1) = \frac{q^o(2)-q^o(1)}{\Delta t}$ correspond to the initial state of the 2nd order dynamical system without any contradiction. This approach corresponds to the physical interpretation of the forward differences as follows: the second time-derivative of a physical quantity at time instant t will determine the future value of the first time-derivative as the present value of the first time-derivative will determine the value of the coordinate in the next future grid point. By a convenient use of little redundancy in the memory usage by maintaining grid values for q, $\dot{q}$, and $\ddot{q}$, the cost function can be easily formulated by incorporating the dynamic model in it.

Evidently, this problem can be tackled by the simple reduced gradient algorithm using the **independent variables** $\{F(1), \dots, F(HL-2)\}$, and the $q^o(1)$, $q^o(2)$, and $\dot{q}^o(1)$ (redundantly coded) **initial conditions**.

3.3 Further Modifications in the Original RHC Approach

Independently of using or evading the use of Lagrange multipliers, the main feature of the RHC controllers is that for the compensation of the effects of modeling errors, they need frequent measurement or estimation of the actual state. Originally, only one step is realized following the exertion of the optimized control force, and the whole next horizon is redesigned from the actually observed new initial state. In Hamza Khan's original adaptive solution it was suggested to use only a few points **for tracking the optimized trajectory (more than one) instead exerting the optimized forces** before making the new observations [94].

A possible new way for dealing with the state observations can be designed in the following manner: instead interrupting the optimization process with observations, it is carried out by the use of a *hypothetical* (but more or less realistic) dynamic model, and instead making measurements, its known computed values can be used in the initial conditions for the next horizon. It is reasonable to assume that the physical system that exactly has this model can move along this optimized trajectory if it obtains the appropriate control forces. Then the **actually controlled system is driven along this optimized trajectory instead of the nominal one** by the adaptive controller that makes the necessary observations and calculates the necessary control forces for the actual dynamic model. If the actual dynamic model is not very far from the assumed approximate version, this adaptive motion can be hopefully realized. If possible, it is expedient to make the optimization for a little bit more heavy dynamic model for which the force limitations may come into effect. In the forthcoming investigation this idea is applied, too.

3.3.1 Simulation Examples for a Furuta Pendulum

In this section this approach is exemplified in the optimal adaptive control of a Furuta pendulum that is a popular paradigm due to its nonlinear dynamic model. The development of the precise dynamic model obtained considerable attention in the literature (e.g., [60, 95–97]).

A dynamic model of strongly overestimated inertia and friction parameters is used in an RHC controller to track the nominal trajectory under cost terms penalizing the control forces. The so obtained optimized trajectory is tracked by an adaptive controller that uses a more realistic approximate dynamic model of the controlled system. Since the approximate and the actual models contain smaller inertia and friction parameters than that used for optimization, it is expected that the cautiously optimized trajectory can be precisely tracked by the actual system without suffering from heavy force burdens.

The *Furuta pendulum* that was invented by *K. Furuta* and his colleagues in 1992 is outlined in Fig. 3.4. It consists of a first arm with m_1 mass and length L_1 in which the distance l_1 refers to location of the center of mass (measured from the rotational axis of the arm). In the same manner, the parameters m_2, L_2 and l_2 are related to the second arm. Parameters $J_{2_{xx}}$,$J_{2_{yy}}$ and $J_{2_{zz}}$ correspond to the moment of inertia of arm 2 around the x-axis, y-axis and z-axis, respectively. However, the first arm is rotating only around axis z so the only moment of inertia of arm-1 is taken with respect to axis z as $J_{1_{zz}}$. The generated torque F_1 and F_2 of DC motors rotate the arms with angle q_1 and angular velocity $\dot{q}_1$ with $\ddot{q}_1$ and arm-2 with angle q_2 and angular velocity $\dot{q}_2$ with $\ddot{q}_2$ joint coordinate acceleration. Parameter b_1 denotes the damping coefficients of the joints. The equations of motion can be set as:

$$\begin{bmatrix} \begin{pmatrix} \ddot{q}_1\Big(J_{1_{zz}}+m_1l_1^2+m_2L_1^2+\left(J_{2_{yy}}+m_2l_2^2\right) \\ \times\sin^2 q_2+J_{2_{xx}}\cos^2 q_2\Big)+\ddot{q}_2m_2L_1l_2\cos q_2 \\ -m_2L_1l_2\sin q_2\dot{q}_2^2+\dot{q}_1\dot{q}_2\sin 2q_2 \\ \times\left(m_2l_2^2+J_{2_{yy}}-J_{2_{xx}}\right)+b_1\dot{q}_1 \end{pmatrix} \\ \begin{pmatrix} \ddot{q}_1m_2L_1l_2\cos^2 q_2+\ddot{q}_2(m2l_2^2+J_{2_{zz}}) \\ +\frac{1}{2}\dot{q}_1^2\sin 2q_2\Big(-m_2l_2^2-J_{2_{xx}}+J_{2_{yy}}\Big) \\ +b_2\dot{q}_2+gm_2l_2\sin q_2 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}. \tag{3.28}$$

The dynamic parameters of the system are given in Table 3.1, the parameters used for the optimization and the approximate parameters used for tracking the optimized trajectory are given in Table 3.2. The whole simulation was set for 4000 steps and discrete horizon length $HL = 6$. The time-resolution of numerical calculations was $\Delta t = 10^{-3}\,[s]$. The cost functions were defined in (3.29) with the parameters $C_q = 1$, $\Delta q = 10^{-3}\,[rad]$, and $P_q = 3$. This structure guarantees that the small tracking errors are nicely penalized with the usual quadratic terms, but the penalization of higher tracking errors becomes more brutal due to the power term 3. With regard to the control torque components, $C_F = 0$ was set when the control torque components were not penalized and $C_F = 1000$ for the penalization of the great control torque components with $\Delta F = 0.001\,[N \cdot m]$ and $P_F = 1$. The gradient descent method was set as follows: $F(n+1) = F(n) - \alpha \nabla \Phi(F(n))$ with $\alpha_{max} = 10^{-8}$ in maximum 1000 steps. If it was necessary, as the minimum was approached, α was decreased to achieve finer localization of the $\|\nabla \Phi\| = 0$ point until either the minimal value $\alpha_{max}/100$ or the maximum step number 1000 was achieved.

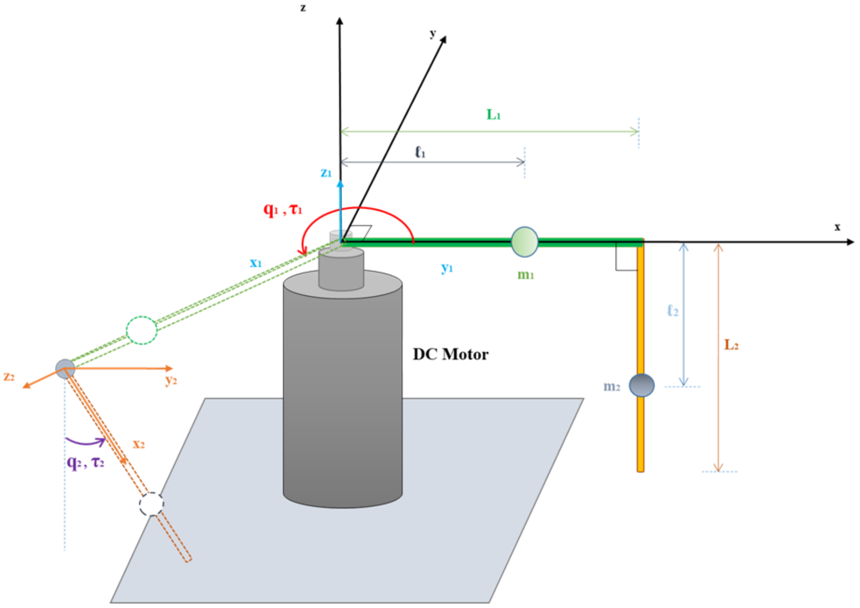

Figure 3.4: Furuta Pendulum ($\tau_1 \equiv F_1$, $\tau_2 \equiv F_2$) on the basis of [97]

Table 3.1: The dynamic model parameters of the pendulum (from [97])

Parameter	Exact Value
Viscous damping coeff b_1 $[N \cdot m \cdot s]$	10^{-4}
Viscous damping coeff b_2 $[N \cdot m \cdot s]$	2.8×10^{-4}
Mass of Arm 1 m_1 $[kg]$	0.300
Mass of Arm 2 m_2 $[kg]$	0.24
Length of Arm 1 L_1 $[m]$	0.278
Length of Arm 2 L_2 $[m]$	0.300
Distance l_1 $[m]$	0.150
Distance l_2 $[m]$	0.148
Moment of inertia $J_{1_{zz}}$ $[kg \cdot m^2]$	1.3×10^{-2}
Moment of inertia $J_{2_{xx}}$ $[kg \cdot m^2]$	1.2×10^{-2}
Moment of inertia $J_{2_{yy}}$ $[kg \cdot m^2]$	2.48×10^{-2}
Moment of inertia $J_{2_{zz}}$ $[kg \cdot m^2]$	2.48×10^{-2}
Gravity $[g\ m \cdot s^{-2}]$	9.81

Table 3.2: The dynamic model parameters of the pendulum used for optimization and tracking the optimized trajectory expressed by the use of the quantities given in Table 3.1

Parameter	**Optimization**	**Tracking**
Viscous damping coeff. b_1 $[N \cdot m \cdot s]$	$2b_1$	$0.5b_1$
Viscous damping coeff. b_2 $[N \cdot m \cdot s]$	$2b_2$	$0.5b_2$
Mass of Arm 1 m_1 $[kg]$	$2m_1$	$0.5m_1$
Mass of Arm 2 m_2 $[kg]$	$2m_2$	$0.5m_2$
Length of Arm 1 L_1 $[m]$	L_1	L_1
Length of Arm 2 L_2 $[m]$	L_2	L_2
Distance l_1 $[m]$	l_1	l_1
Distance l_2 $[m]$	l_2	l_2
Moment of inertia $J_{1_{zz}}$ $[kg \cdot m^2]$	$2J_{1_{zz}}$	$0.5J_{1_{zz}}$
Moment of inertia $J_{2_{xx}}$ $[kg \cdot m^2]$	$2J_{2_{xx}}$	$0.5J_{2_{xx}}$
Moment of inertia $J_{2_{yy}}$ $[kg \cdot m^2]$	$2J_{2_{yy}}$	$0.5J_{2_{yy}}$
Moment of inertia $J_{2_{zz}}$ $[kg \cdot m^2]$	$2J_{2_{zz}}$	$0.5J_{2_{zz}}$
Gravity g $[m \cdot s^{-2}]$	g	g

$$\begin{aligned}
\Phi(q(\ell),F(\ell)) &= \sum_{j=1}^{2} \Phi_j^q(\ell) \quad + \sum_{j=1}^{2} \Phi_j^F(\ell) \\
\Phi_j^q(\ell) &= C_q \begin{cases} \left|\frac{q_j^N(\ell)-q_j^o(\ell)}{\Delta q}\right|^{P_q} & \text{if } \left|\frac{q_j^N(\ell)-q_j^o(\ell)}{\Delta q}\right| > 1 \\ \left|\frac{q_j^N(\ell)-q_j^o(\ell)}{\Delta q}\right|^{2} & \text{if } \left|\frac{q_j^N(\ell)-q_j^o(\ell)}{\Delta q}\right| \leq 1 \end{cases} \\
\Phi_j^F(\ell) &= C_F \left|\frac{F_j(\ell)}{\Delta F}\right|^{P_F}
\end{aligned} \tag{3.29}$$

The output of simulations were made for non-adaptive and adaptive case under force penalization against the force limitation with adaptivity. Figure. 3.5 describes both the non-adaptive and adaptive tracking with force penalization against the output response of the adaptive controller without force penalization, also it reveals that the force penalization causes observable error in tracking the nominal trajectory. It can be seen that the adaptive solution precisely can track the optimized trajectory.

For the adaptivity, in the case of the Furuta pendulum, I used Dineva's adaptive deformation function given in (3.20) with $F(x) = 0.5x + D$ with $D = 0.3$, $A = -2.0$, and in the kinematic block $\Lambda = 12.0\,[s^{-1}]$ in (3.3).

In Fig. 3.6 the trajectory tracking error between nominal and optimized trajectories while the Fig. 3.7 illustrates the error of tracking of the optimized trajectory (i.e., the difference between the optimized the realized trajectories). The appropriate control torque components are described in Fig. 3.8. It is evident that the torque components that are necessary for tracking the optimized trajectory in their absolute values are much smaller than their counterparts that occurred during the optimization. Figure. 3.9 reveals the operation of the adaptation mechanism: considerable adaptive deformation is required for precise tracking of the optimized trajectory.

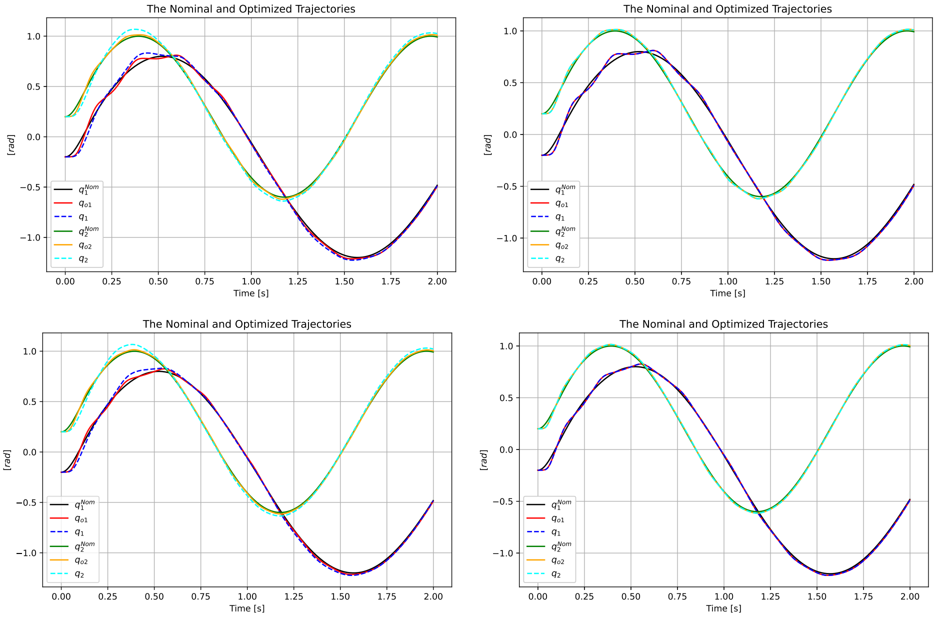

Figure 3.5: At the top: The trajectory tracking of the non-adaptive case (LHS) and in the adaptive case (RHS) with force penalization. At the bottom: output responses without force penalization.

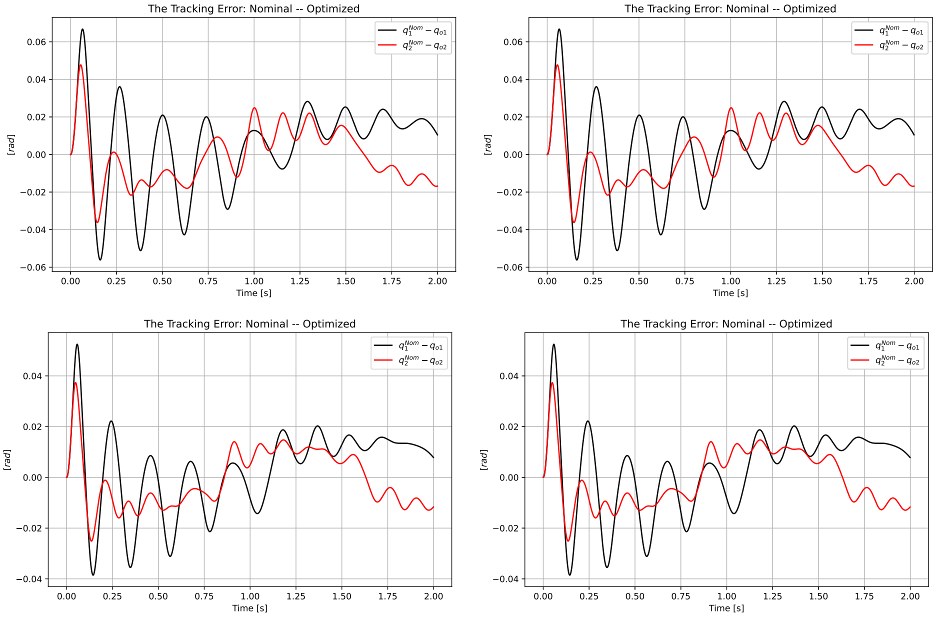

Figure 3.6: At the top: The error of tracking the nominal trajectory by the optimized one, and tracking error between the optimized and the realized trajectories under force penalization (in the adaptive case). At the bottom: output responses without force penalization.

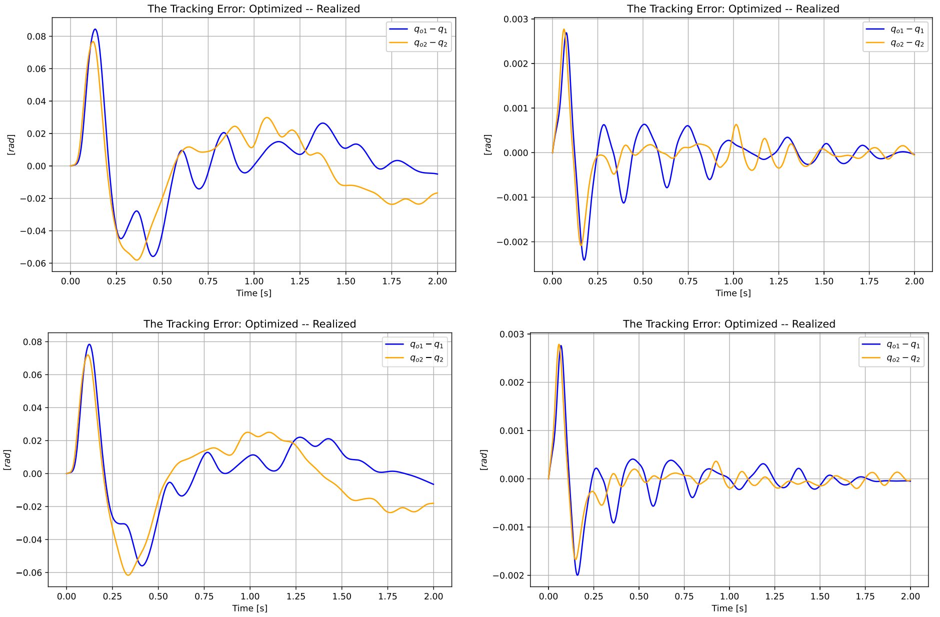

Figure 3.7: At the top: The error of tracking the optimized trajectory in the non-adaptive case (LHS) and in the adaptive case (RHS) under force penalization. At the bottom: output responses without force penalization.

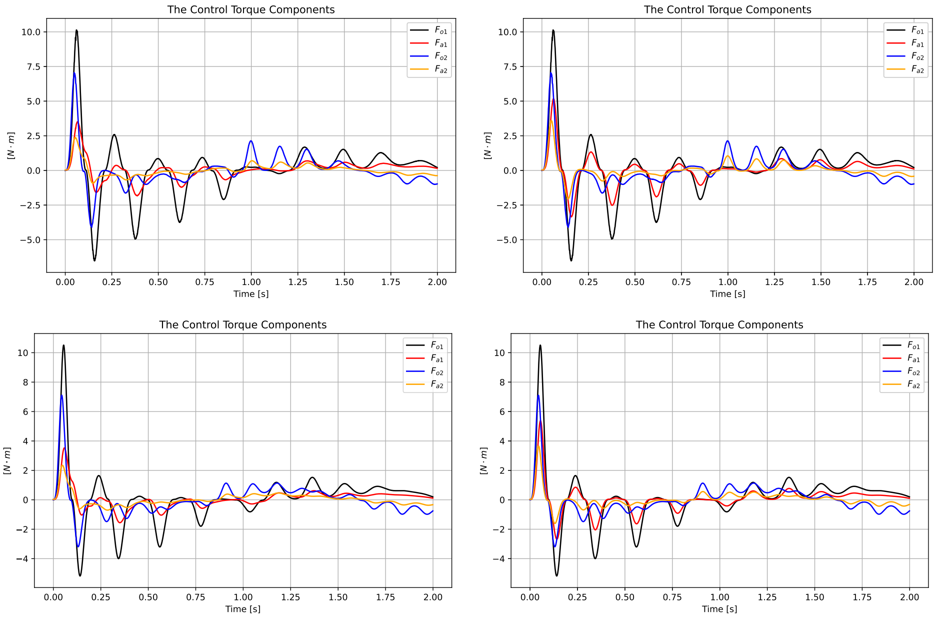

Figure 3.8: At the top: The control torque components F_1 and F_2 in the non-adaptive case (LHS) and in the adaptive case (RHS) with force penalization (F_{o1} and F_{o2} denote the force need of the optimized system with overestimated parameters, F_{a1} and F_{a2} denote the force need of tracking the optimized trajectory by the actual system). At the bottom: output responses without force penalization.

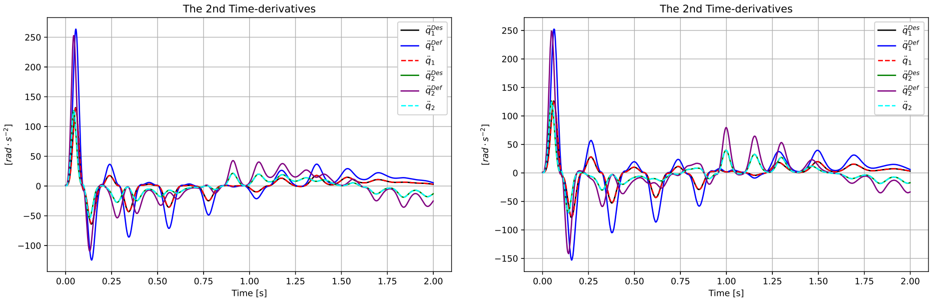

Figure 3.9: The adaptation mechanism without force penalization (LHS) and in the case of force penalization (RHS).

3.3.1.1 Simulations with LuGre Friction Model

In these simulations the particular form of the LuGre friction model was taken from [98]. In this case the parameters given in Table 3.3 were in use. In this model for the description of the phenomenon of stick-slip a hidden dynamic variable, $z(t)$ is introduced that physically is related to the behavior of the bristles of connected brushes. Depending on the velocity of the surfaces in touch $v(t)$, the actual value of $\dot{z}(t)$ is determined by $v(t)$ and $z(t)$, and the friction force $F(t)$ depends on $z(t)$, $\dot{z}(t)$, and $v(t)$ according to (3.30).

$$\frac{dz}{dt} = v - \frac{\sigma_0 |v|}{F_c + F_s exp(-|v|/v_s)} z \tag{3.30a}$$

$$F = \sigma_0 z + \sigma_1 \frac{dz}{dt} + F_v \times v \quad . \tag{3.30b}$$

Table 3.3: The LuGre model parameters of the simulations

Parameter	Value
σ_0 $[Nm^{-1}]$ Stiffness	20.0
σ_1 $[Nsm^{-1}]$ Viscous friction	8.0
F_C $[N]$ Coulomb friction force	2.2
F_S $[N]$ Stribeck friction force	1.1
F_v $[Nsm^{-1}]$ Static friction force	0.3
v_s $[ms^{-1}]$ Characteristic velocity of Stribeck friction	0.5

The cost functions were defined in (3.29) with the parameters $C_q = 5$, $\Delta q = 10^{-5}\,[rad]$, and $P_q = 4$. With regard to the control torque components, $C_F = 2$,$\Delta F = 0.01\,[N \cdot m]$ and $P_F = 1$ were set.

Figures 3.10 and 3.12 reveal that in spite of the drastic not modeled friction forces given in Fig. 3.13, the adaptive tracking was quite precise. It can be seen that where great negative friction forces occurred in time, the control forces have been increased, and vice versa. The phase trajectories of the hidden internal variables of the friction model in Fig. 3.13 well represent the stick-slip phenomenon captured by the LuGre model. The graphs of the second time-derivatives of the joint coordinates in Fig. 3.11 display the great extent of the necessary adaptive deformation required for defying the friction effects not modeled in the optimization. The control parameters were in $F(x) = 0.5x + D$ with $D = 0.15$ and $A = -5.0$ in (3.20), and $\Lambda = 15.0\,[s^{-1}]$ in (3.3).

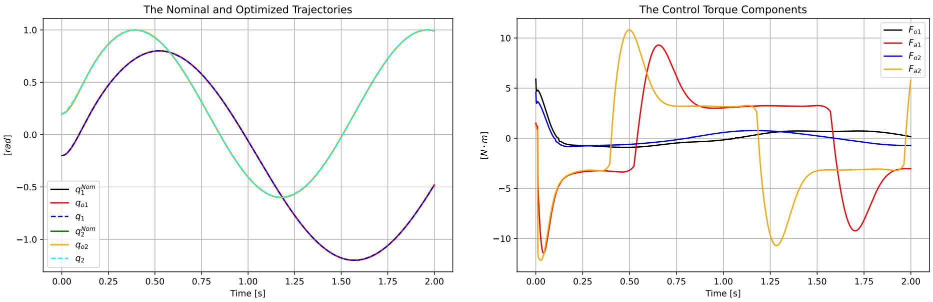

Figure 3.10: The trajectory tracking (LHS) and generated forces (RHS)

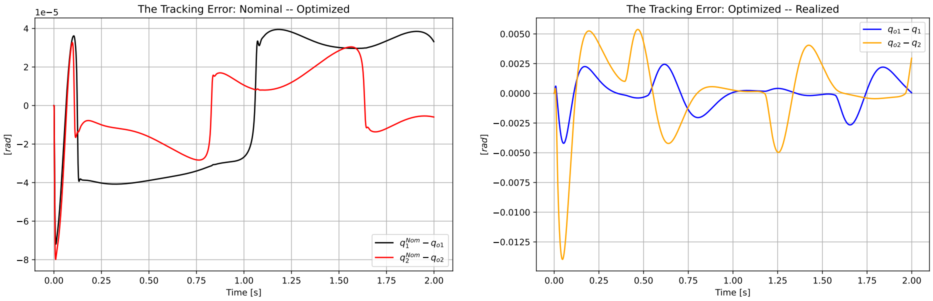

Figure 3.11: The trajectory tracking error for nominal and optimized (LHS) and trajectory tracking error for optimized and realized.

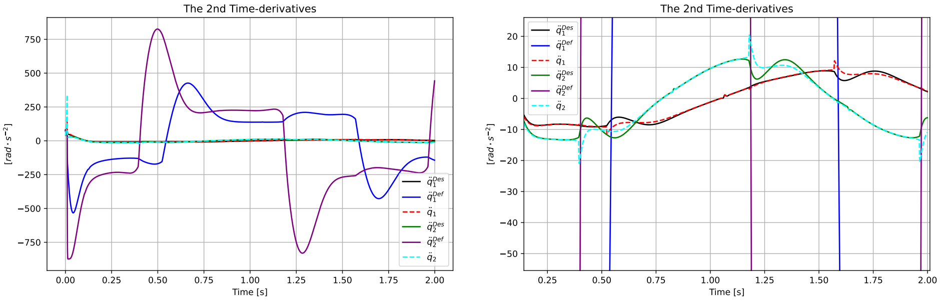

Figure 3.12: The accelerations (LHS) and its zoomed in excerpt (RHS)

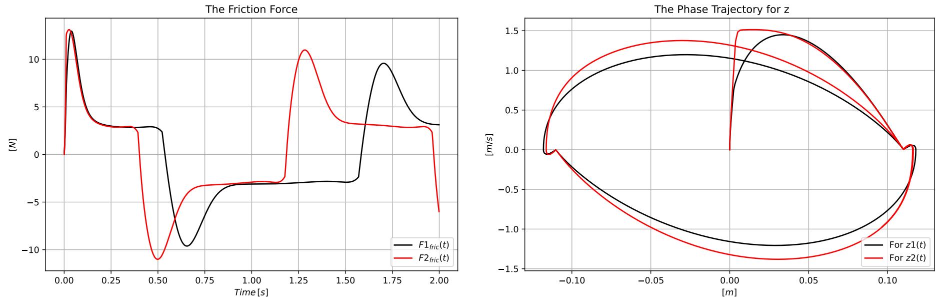

Figure 3.13: The Friction Force (LHS) and the Phase trajectory for z (RHS)

3.3.2 Design of Adaptive Receding Horizon for SCARA Robot

In this session the idea that was at first considered for the control of the Furuta pendulum, was further tested for a different physical system, the 4-DOF SCARA robot. In the suggested solution the dynamic model is directly used without any gradient reduction by using a transition between the gradient descent and the Newton–Raphson methods to achieve possibly fast operation. The optimization is carried out for an overestimated dynamic model, and instead of using the optimized force components the optimized trajectory is adaptively tracked by an available approximate dynamic model of the controlled system. Also, the dynamic model was built in the cost function.

The speed of optimization certainly strongly depends on the properties of the cost functions in use. In [77, 78] very complex cost functions were introduced that were based on polynomial behavior of the functions $\left(\frac{|x|}{\Delta}\right)^p$. If $p > 1$ such functions have little contribution if $|x| \ll \Delta > 0$ (this is an error tolerant region), and drastically increase as $|x| \gg \Delta$ (it corresponds to a strongly penalized region). However, numerical problems often arose.

In the present investigations, the cost function was composed for the generalized coordinate q_j for penalizing tracking error and the penalization of the driving force u_j term as in (3.29) as follows:

$$\Phi_j^q(\ell) := C_{q_j} \frac{\left|(q_j^N(t) - q_j^O(t))/\delta\right|^{p+1}}{1 + \left|(q_j^N(t) - q_j^O(t))/\Delta\right|^p} \;, \Phi_j^F(\ell) = C_{u_j} \begin{cases} |F_j - \Delta_u|^{p_u} \text{ if } F_j > \Delta_u \\ 0 \text{ if } -\Delta_u \le F_j \le \Delta_u \\ |F_j + \Delta_u|^{p_u} \text{ if } F_j < -\Delta_u \end{cases} \tag{3.31}$$

For the big tracking error, its graph looks like that of a linear function to evade numerical problems, whereas for small errors it is tolerantly flat and small (typical shapes are given in Fig. 3.14). With regard to the force penalization, it can be stated that a range of forces were exerted without any penalization.

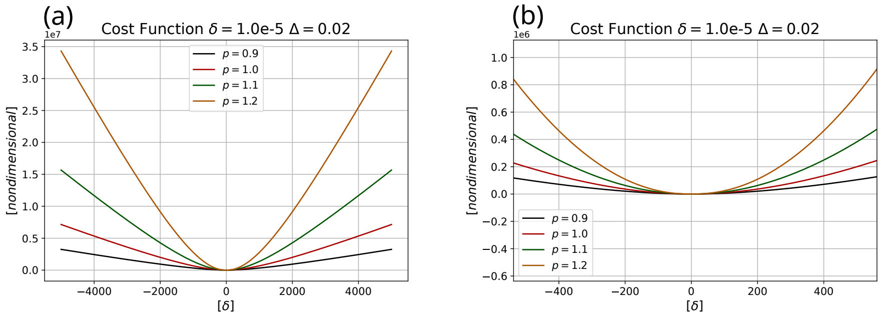

Figure 3.14: The schematic structure of the cost function shape for penalization (for arbitrary physical dimensions) $\delta = 10^{-5}$, $\Delta = 2000\delta$, and various p values: (**a**) in full scale; (**b**) zoomed in excerpt.

In this manner, a relatively fast algorithm was created for seeking the possible minimum of the cost function, because the linear part gives very fast convergence in the Newton–Raphson algorithm. Figure. 3.14 reveals, that this algorithm must slow down at the flat parts belonging to small errors.

For the optimization with a heavy model (i.e., a model having bigger inertia data and assuming higher gravitational acceleration as the realistic one), and adaptively tracking the optimized trajectory with a more realistic model, numerical data given in Table 3.4 were used. (The length data are identical for the heavy and the exact models, but the masses, the inertia moments, and the gravitational acceleration are overestimated for the heavy model.)

Therefore, the force values that were calculated in the optimization process were dropped and the smoothed version $q^{Os}(t)$ of the optimized trajectory $q^{O}(t)$ was adaptively tracked by the use of an available approximate dynamic model of the actual system under control. For smoothing, a third-order solution inspired by [99] was applied as

$$\left(\Lambda_{filt}+\frac{\mathrm{d}}{\mathrm{d}t}\right)^3 q^{Os}(t)=\Lambda_{filt}^3 q^{O}(t)\,,\; q^{Os}(t_0)=0\,,\; \dot{q}^{Os}(t_0)=0\,,\; \ddot{q}^{Os}(t_0)=0\,, \tag{3.32}$$

with $0<\Lambda_{filt}=\text{constant}$. For very high frequencies it has the transfer function characteristic in the Laplace transform as $\propto s^{-3}$ (very drastic rejection), and at zero frequency it has the value 1.

The whole control process for having each optimal coordinate q_i^O, filtered optimal coordinate q_i^{Os} and the realized one q_i can be recognized through the flow chart in Fig. 3.15.

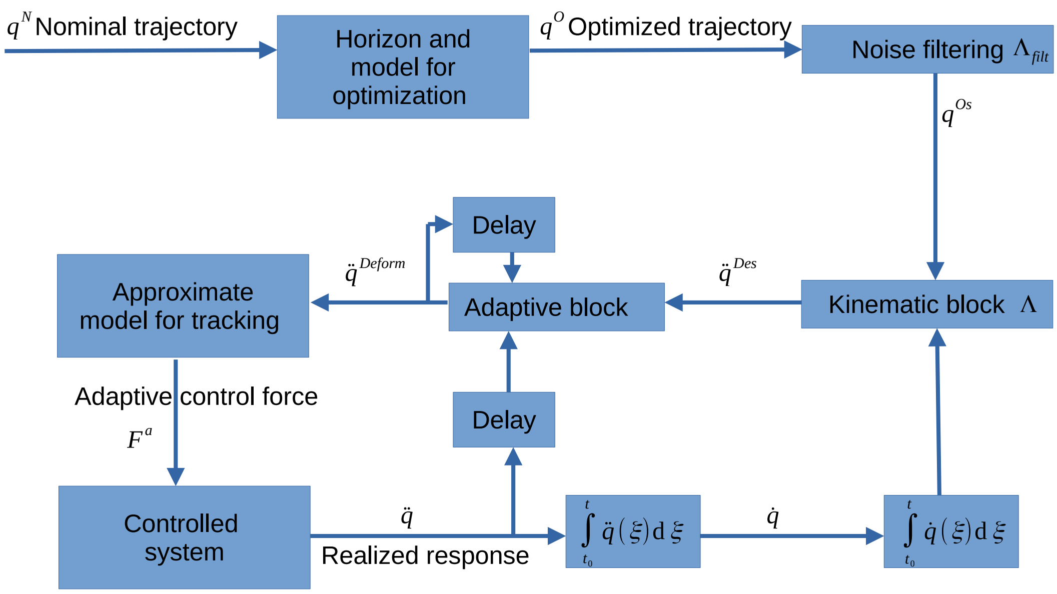
q^N Nominal trajectory
Horizon and model for optimization
q^O Optimized trajectory
Noise filtering Λ_{filt}
q^{Os}
Delay
$\ddot{q}^{Deform}$
$\ddot{q}^{Des}$
Approximate model for tracking
Adaptive block
Kinematic block Λ
Adaptive control force
F^a
Delay
Controlled system
$\ddot{q}$
Realized response
$\int_{t_0}^{t} \ddot{q}(\xi)\,d\xi$
$\dot{q}$
$\int_{t_0}^{t} \dot{q}(\xi)\,d\xi$

Figure 3.15: The flow chart of the controlling mechanism.

The dynamic model of the 4-DOF SCARA robot arm was taken from [100]. The generalized coordinates of the robots are q_1 [m] is the only prismatic joint, and q_2, q_3, q_4 are rotary ones measured in [rad] units. Accordingly, the generalized forces are F_1 [N] for the first joint, and F_2, F_3, F_4 have the dimension [N · m]. The equations of motion are given in (3.33)–(3.36),

$$F_1 = (m_1 + m_2 + M + M_{load})\ddot{q}_1 - g(M + m_1 + m_2 + M_{load}) \ , \tag{3.33}$$

$$\begin{aligned} F_2 = {} & (m_1 L_1^2/4 + m_2 L_1^2 + m_2 L_2^2/2 + m_2 L_1 L_2 \cos(q_3) + M_{load}(L_1^2 + L_2^2 + 2L_1 L_2) + \Theta_{load})\ddot{q}_2 \\ & + (m_2 L_2^2/2 + m_2 L_1 L_2 \cos(q_3)/2 + M_{load}(L_2^2 + L_1 L_2 \cos(q_3)) + \Theta_{load})\ddot{q}_3 + \Theta_{load}\ddot{q}_4 \\ & - \dot{q}_2(m_2 L_1 L_2 \sin(q_3) + M_{load} 2 L_1 L_2 \sin(q_3))\dot{q}_3 - \dot{q}_3^2(m_2 L_1 L_2 \sin(q_3) + M_{load} L_1 L_2 \sin(q_3)) \ , \end{aligned} \tag{3.34}$$

$$\begin{aligned} F_3 = {} & (m_2 L_2^2/2 + m_2 L_1 L_2 \cos(q_3)/2 + M_{load}(L_2^2 + L_1 L_2 \cos(q_3)) + \Theta_{load})\ddot{q}_2 \\ & + (m_2 L_2^2/4 + M_{load} L_2^2 + \Theta_{load})\ddot{q}_3 + \Theta_{load}\ddot{q}_4 + \dot{q}_2^2 L_1 L_2 sin(q_3)(m_2/2 + M_{load}) \ , \end{aligned} \tag{3.35}$$

$$F_4 = \Theta_{load}\ddot{q}_2 + \Theta_{load}\ddot{q}_3 + \Theta_{load}\ddot{q}_4 \ . \tag{3.36}$$

The dynamic parameters of the heavy model used for optimization, the exact system model (not known by the controller), and the available approximate system model used for trajectory tracking are given in Table 3.4.

Table 3.4: The Dynamic Model Parameters in Equations (3.33)–(3.36).

Parameter	Exact Model for Simulation	Heavy Model for Optimization	Approximate Model for Adaptive Control
M[kg] component's mass	10.0	15.0	12.0
m_1[kg] component's mass	20.0	25.0	21.0
m_2[kg] component's mass	10.0	13.0	12.0
M_{load}[kg] load's mass	50.0	55.0	52.0
g [m $\cdot$ s^{-2}] grav. accel.	9.81	10.0	9.0
θ_{load} [kg $\cdot$ m^2] load's inertia moment	45.0	50.0	42.0
L_1[m] arm length	2.0	2.0	2.0
L_2[m] arm length	1.0	1.0	1.0

3.3.2.1 Simulation Results

The necessary time-grid resolution depends on various factors such as the dynamics of the nominal trajectory to be tracked, the structure of the cost function applied, the parameters used in smoothing the optimized trajectory, and that of the adaptive controller that tracks the smoothed trajectory. From a practical point of view, making numerical simulations for a given problem seems to be an easier way. A simple check of reliability is the comparison of the results obtained for the sets $\{\delta t = 10^{-4}\,[s], STEPS = 4000, HL = 12\}$ (referred to as set 1) and $\{0.5\delta t, 2 \cdot STEPS, 2 \cdot HL\}$ (referred to as set 2) that physically corresponds to computing the same task with a finer time resolution. If the results obtained for the tracking precision and the control force needs can be well compared to each other, the original time resolution δt can be considered as acceptable. Table 3.5 shows the controller parameter values that used during all the simulation parts. The tracking parameter must be great enough to track the dynamic of the optimized motion, and to evade strong noise-sensitivity it is not expedient to use too much value. The smoothing parameter must be relatively large in order to follow the fast correction terms to the optimized trajectory's motions, but no much higher value is needed because its increase also increases the role of the higher frequency noise components. For the particular simulations these parameters were set experimentally. In general they depend on the needs of the particular application.

Table 3.5: The Controller's Parameters.

Parameter	***Meaning***	***Value***
δt	Discrete time resolution	10^{-4}[s]
Λ	Trajectory tracking exponential coeff.	36.0 [s^{-1}]
Λ_{filt}	Trajectory smoothing exponential coeff.	800.0 [s^{-1}]
$C_{q_1} = C_{q_3} = C_{q_4}$	Cost contribution coeffs.	10^6
C_{q_2}	Cost contribution coeff.	2×10^6
δ	Cost parameter 1	10^{-5} [rad] or [m]
Δ	Cost parameter 2	$2000 \cdot \delta$
$C_{u_1} = C_{u_2} = C_{u_3} = C_{u_4}$	Force cost parameter 1	10^4
p	Cost parameter 3	1.1
Δ_u	Force cost parameter 2	varying [$N \cdot m$] or [N]
p_u	Force cost parameter 3	1.5
R_a	Augmented arrays' Frobenius norm	10^6 [m $\cdot$ s^{-2}] or [rad $\cdot$ s^{-2}]
H	Discrete horizon length	12
$\ddot{q}_{maxa}$	Moderating factor in adaptive control	10^4 [m $\cdot$ s^{-2}] or [rad $\cdot$ s^{-2}]
$\ddot{q}_{maxk}$	Moderating factor in kinematic block	10^7 [m $\cdot$ s^{-2}] or [rad $\cdot$ s^{-2}]
α_{min}	Stopping limit in minimum seeking	10^{-2}
λ_a	Adaptive interpolation factor	1.0

I. Simulations Without Force Limitation – Different Time Resolutions

In the simulation investigations, the operation of the method at first was considered for a cost function without control force penalizing terms (in this case $\forall j C_{u_j} = 0$). The common control parameters of these simulations are given in Table 3.5. Some sinusoidal motion was chosen for the nominal trajectory to be tracked.

It can be seen in Figs. 3.16–3.19 that halving the time-resolution $\delta t = 10^{-4}\,[s]$ did not produce significantly different results. The adaptive abstract rotations have no noticeable differences (Fig. 3.20). Naturally, the computational burden of the method strongly depends on the time resolution which can be noticed in the big differences of the computational time need in Fig. 3.21. On this basis, it was determined that for a preliminary design the computationally less greedy set 1 parameters' setting will be used in the case of force limitations. It worth mentioning that the big initial transient is a typical consequence of a PID-type tracking policy.

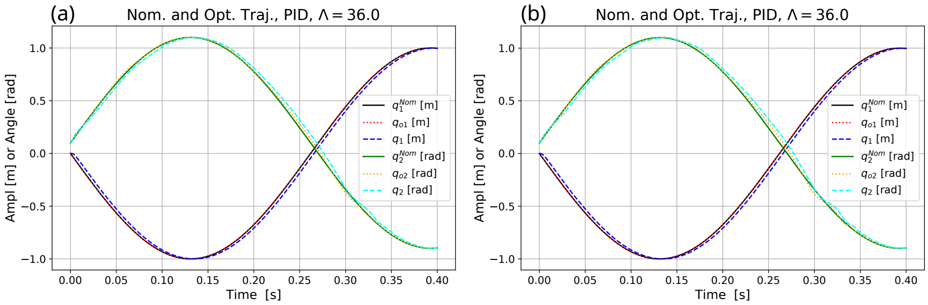

Figure 3.16: Trajectory tracking without force limitation for q_1 and q_2: (**a**) For the set 1. (**b**) For the set 2.

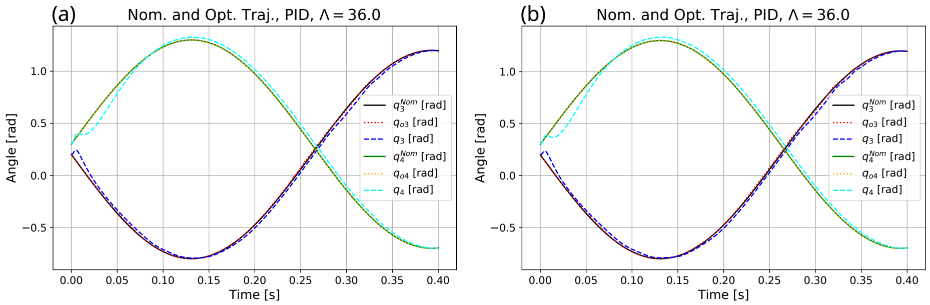

Figure 3.17: Trajectory tracking without force limitation for q_3 and q_4. (**a**) For the set 1. (**b**) For the set 2.

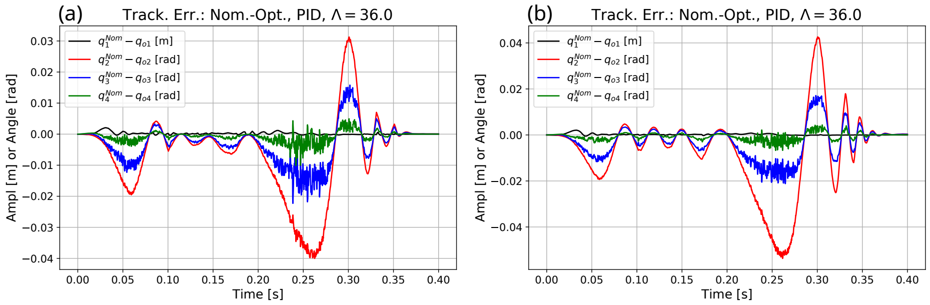

Figure 3.18: Nominal optimized trajectory tracking errors $q^N - q^O$ without force limitation. (**a**) For set 1.(**b**) For set 2.

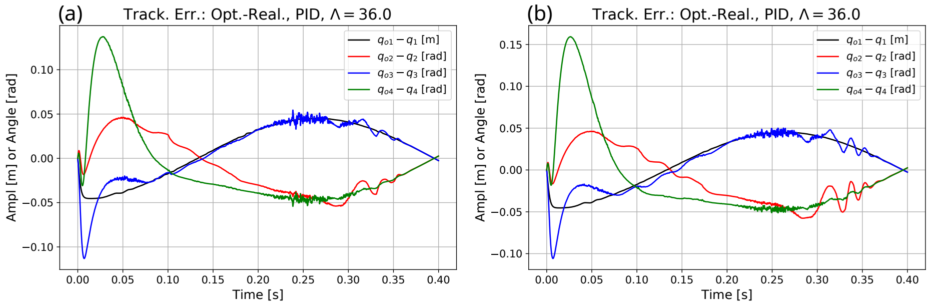

Figure 3.19: Optimized realized trajectory tracking errors $q^O - q$ without force limitation. (**a**) For set 1. (**b**) For set 2.

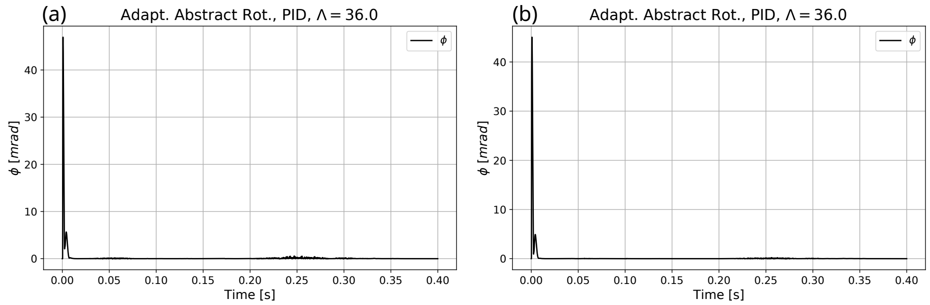

Figure 3.20: The adaptive abstract rotations. (**a**) For set 1. (**b**) For set 2.

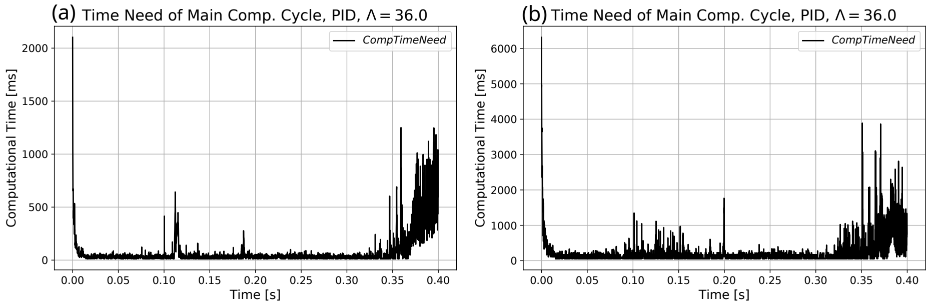

Figure 3.21: The computational time need of the main cycle without force limitation. (**a**) For set 1. (**b**) For set 2.

II. Simulations Without Force Limitation – With Higher Minimum Error Stopping Conditions

Based on previous comparison, set 1 will be chosen for the upcoming investigations. In the sequel, to decrease computational burden, the setting $\alpha_{min} = 10^{-1}$ value was used. This corresponds to a less precise approximation of the local minimum, but the noise filtering applied for adaptive tracking can tackle the effects of this increased imprecision. According to Figs. 3.22 and 3.23, it can be stated that the RHC controller was able to generate a good optimized trajectory that was successfully tracked by the adaptive controller. The comparison of Figs. 3.21 and 3.24b reveals quite considerable reduction in the necessary computational time. (Instead of counting the necessary mathematical operations for estimating the hypothetical time need for the goal of comparison, the time needs were measured by the use of a given hardware and software system, in this case the service of the Julia language. The actual times indicate that this hardware is not appropriate for realizing the control that is needed for such fast robot motion. However, it satisfactory for making comparisons. In the practice the available hardware and software tools determine the achievable computational speed. The computer must be fast enough to solve a given control task. For the solution of the same task various mathematical solutions exist. Those solutions that use simpler and faster mathematical operations for solving a given problem can spare time. Therefore, they can control faster motions than the methods that need more complex and resource greedy mathematical operations.) The adaptive abstract rotation is within accepted range as it can be seen in Fig. 3.24a.

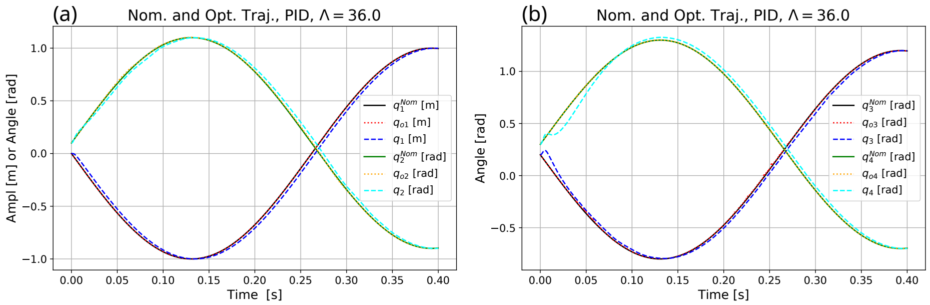

Figure 3.22: Trajectory tracking without force limitation for the increased $\alpha_{min} = 0.1$. (**a**) For q_1 and q_2. (**b**) For q_3 and q_4.

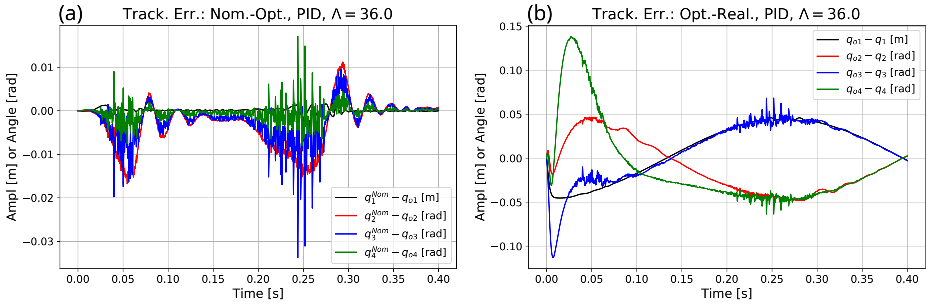

Figure 3.23: Trajectory tracking errors without force limitation for the increased $\alpha_{min} = 0.1$. (**a**) For nominal-optimized: $q^N - q^O$. (**b**) For optimized-realized: $q^O - q$.

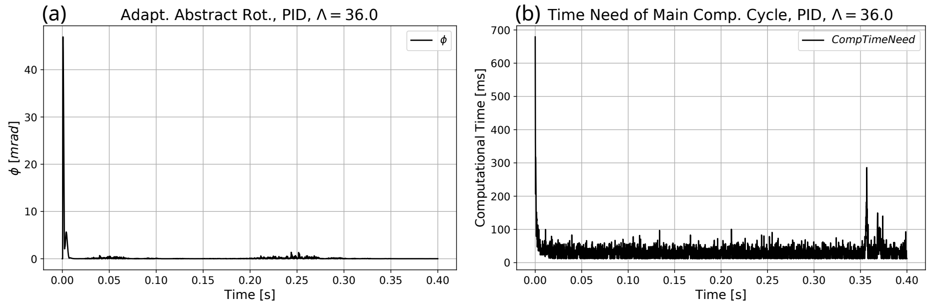

Figure 3.24: The adaptive abstract rotations (**a**) and the computational time-need of the main cycle without force limitation (**b**) for the increased $\alpha_{min} = 0.1$.

III. Simulations With Force Limitation

These simulations were made by keeping the increased $\alpha_{min} = 0.1$ and by choosing the set 1; otherwise the parameters given in Table 3.5 were in use. The value of the parameter $\Delta_u = 495.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ considerably corrupted the optimized trajectory by not allowing it to exert the necessary control forces. Figures. 3.25 and 3.26 reveal that in the optimization phase the force limitation seriously concerned the optimized trajectories $q_1^O(t)$, $q_2^O(t)$,

$q_3^O(t)$, and $q_4^O(t)$, the smoothed versions of which were adaptively well tracked.According to Fig. 3.27b the time need of the optimum seeking was kept at a relatively low level.

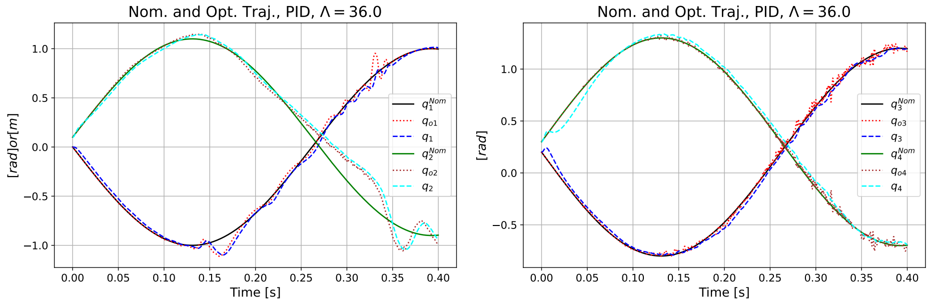

Figure 3.25: Trajectory tracking with force limitation $\Delta_u = 495.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ for the increased $\alpha_{min} = 0.1$. (**a**) For q_1 and q_2. (**b**) For q_3 and q_4.

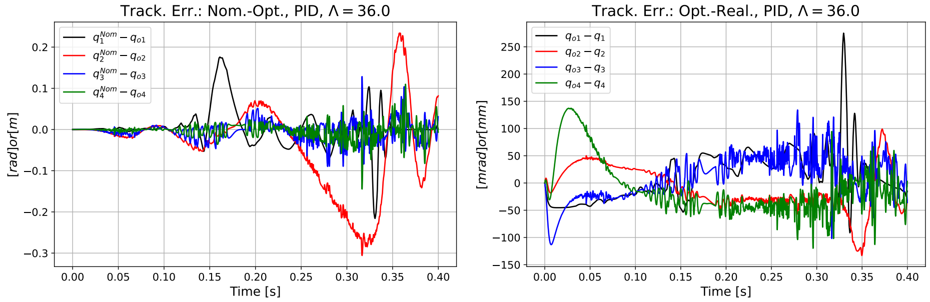

Figure 3.26: Trajectory tracking errors with force limitation $\Delta_u = 495.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ for the increased $\alpha_{min} = 0.1$: (**a**) For nominal-optimized: $q^N - q^O$. (**b**) For optimized-realized: $q^O - q$.

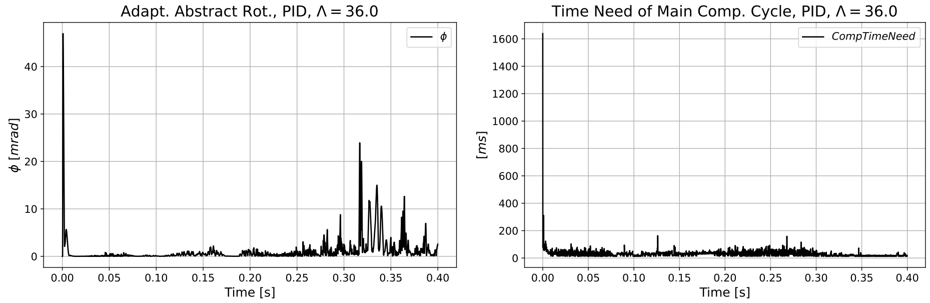

Figure 3.27: The adaptive abstract rotations (**a**) and the computational time need (**b**) of the main cycle with force limitation $\Delta_u = 495.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ for the increased $\alpha_{min} = 0.1$.

It is interesting to observe how sharply the parameter value Δ_u concerns the results for $p_u = 1.5$. If the limits of the penalty-free force region are increased from $\Delta_u = 495.0$ to $\Delta_u = 500.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ the optimized trajectory suffered far fewer distortions (Fig. 3.28).

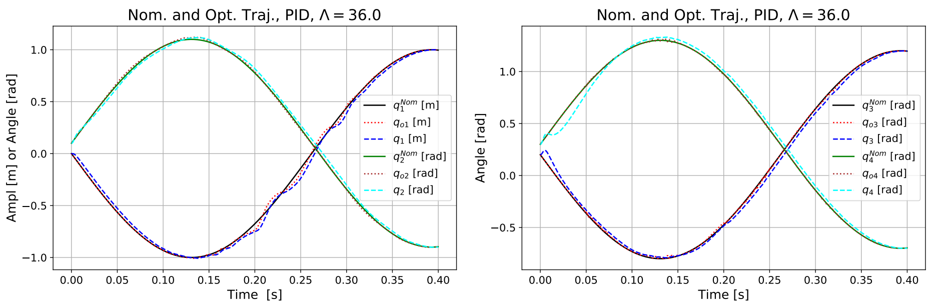

Figure 3.28: Trajectory tracking with force limitation $\Delta_u = 500.0\,[N]$ or $[\mathrm{N}\cdot\mathrm{m}]$ for the increased $\alpha_{min} = 0.1$: (**a**) For q_1 and q_2. (**b**) For q_3 and q_4.

3.4 General Receding Horizon Solution of the Inverse Kinematic Task

A quasi-differential approach was elaborated for solving the inverse kinematic task of robots in Section 2.2 that, though calculates the Jacobian, instead inverting it, applies a fixed point iteration that automatically evades the singularities. However, its solution depends on the eigenvalues of the Jacobian, and it was found not flexible enough for all practical purposes. The Moore-Penrose solution can be generalized as an optimization task under constraints for quite complex cost functions. The hard constraint in this approach forces the exact solution of the inverse kinematic task when it is possible. Due to it the intent of minimizing the costs is overridden in the vicinity of the singularities. In the present approach, based on the formal structure of the Receding Horizon Controllers, an alternative solution is suggested that allows flexibility for relaxing the costs and the hard constraints, too. Its computational complexity to some extent is reduced via evading the technique of gradient reduction. Simulation results are presented for the same 7-DOF robot arm that was considered in Section 2.2. In the selected solutions the $\dot{q}$ variables were treated as the independent variables of the optimization (instead of the force terms of the *dynamical problems*), and the q coordinates were calculated as dependent ones in a little bit redundant representation as follows:

- The Nominal Cartesian Trajectories $X^N(1),\dots,X^N(HL)$ that are known in advance since definite ideas are available for the future of the nominal motion are given over the horizon;

- The also known initial joint coordinates in the first grid point $q(1)$ taken from the last actual value (in the control applications it is measured or somehow observed);

- By using the $\dot{q}$ values in the grid points $\{\dot{q}(1),\dots,\dot{q}(HL-1)\}$, with Euler integration, fill in the horizon points as $q(i+1) = q(i) + \Delta t\dot{q}(i)$ for $i = 1,\dots,HL-1$;

- By the use of the *model function* compute the *assumed future Cartesian values* $X^O(i) = F\left(q(i)\right)$ for $i \in \{2,\dots,HL\}$;

- Compute the cost functions over the grid points $i \in \{2,\ldots,HL\}$ and the penalty values for the $\dot{q}$ components over the grid points $i \in \{1,\ldots,HL-1\}$ as

$$\Phi = \sum_{j=1,i=2}^{6,HL} \Phi_j^X(i) \; + \sum_{j=1,i=1}^{6,HL-1} \Phi_j^U(i)$$

$$\Phi_j^X(i) = \begin{cases} \left|\frac{X_j^N(i)-X_j^O(i)}{\Delta x}\right|^{P_X} & \text{if } \left|\frac{X_j^N(i)-X_j^O(i)}{\Delta x}\right| > 1 \\ \left|\frac{X_j^N(i)-Xj^O(i)}{\Delta x}\right|^2 & \text{if } \left|\frac{X_j^N(i)-X_j^O(i)}{\Delta x}\right| \leq 1 \end{cases}$$

$$\Phi^U(i) = \begin{cases} \left|\frac{\dot{q}(i)}{\Delta \dot{q}}\right|^{P_{\dot{q}_{out}}} & \text{if } \left|\frac{\dot{q}(i)}{\Delta \dot{q}}\right| > 1 \\ \left|\frac{\dot{q}(i)}{\Delta \dot{q}}\right|^{P_{\dot{q}_{in}}} & \text{if } \left|\frac{\dot{q}(i)}{\Delta \dot{q}}\right| \leq 1 \end{cases}$$

- The so computed cost function can be minimized by using the simple *Gradient Descent Algorithm* so that the components of $\nabla\Phi$ can be arranged in a matrix of size $\mathbb{R}^{6\times(HL-1)}$;

- The starting point of the next horizon will be the element $q(2)$ of the so optimized horizon, and the joint coordinates' time-derivatives can be estimated by the value that is found in $\dot{q}(1)$ of the present horizon after the optimization.

In the cost contribution $\Phi_j^X(i)$ the component tracking errors that in absolute value are greater than Δx are very strongly penalized if $P_X > 1$, and for smaller error components the usual quadratic tracking rule is prescribed. In similar manner, in the term $\Phi_j^U(i)$, the joint coordinate time-derivatives are very seriously penalized if their absolute value is bigger than $\Delta\dot{q}$ and $P_{\dot{q}_{out}} > 1$, but the small values' penalty contribution is very small if $P_{\dot{q}_{in}} > 1$. Therefore it is expected that by the use of these shape parameters the trajectory tracking precision can be relaxed if the high $\dot{q}$ values are seriously penalized.

For describing the orientation and its error, in these simulations it was taken into account that $G^T = -G$ and $G^{2T} = G^2$ the even power components in (3.37) are *symmetric* while the odd power ones are *skew symmetric*, the very redundant representation of O, in which 9 matrix elements are used for describing the rotations of only 3 independent parameters can be reduced if only the three independent components of the skew symmetric part of O, namely $O^A := \left(O - O^T\right)/2$ are considered for expressing the pose of the tool, as e.g., O^A_{12}, O^A_{13}, and O^A_{23}. On this basis a *model function* $F : \mathbb{R}^n \mapsto \mathbb{R}^6$ is used for describing the forward kinematic task as $F_1(q_1,\ldots,q_n) = O^A_{12}$, $F_2(q_1,\ldots,q_n) = O^A_{13}$, $F_3(q_1,\ldots,q_n) = O^A_{23}$, $F_4(q_1,\ldots,q_n) = r_1$, $F_5(q_1,\ldots,q_n) = r_2$, and $F_6(q_1,\ldots,q_n) = r_3$ as is given in (3.37).

$$\begin{aligned} O = \exp(qG) = I + &\begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \sin q + \\ + &\begin{bmatrix} e_1^2 - 1 & e_1 e_2 & e_1 e_3 \\ e_2 e_1 & e_2^2 - 1 & e_2 e_3 \\ e_3 e_1 & e_3 e_2 & e_3^2 - 1 \end{bmatrix} (1 - \cos q) \; . \end{aligned} \tag{3.37}$$

3.4.1 Simulation Results

In the 7-DOF redundant robot system described in Fig. 2.10 the homogeneous coordinates related to the TCP were $\tilde{r} = [1.0, 0, 0, 1]$. The other kinematic parameters are given in Table 3.6.

Table 3.6: The parameters of the robot model

Parameters in Fig. 2.10	**Value**
The height of the 1^{st} horizontal axle h [m]	1.0
For the 1^{st} long arm S [m]	2
For the 2^{nd} long arm l [m]	1
For the prismatic arm L [m]	0.5

The time-resolution was $\Delta t = 10^{-3}\,[s]$, the horizon length was $HL = 3$, the algorithm was stopped when the situation $\|\nabla\Phi\| \leq \mu = 10^{-4}$ was achieved, the components of the gradient were estimated with a final step length $\delta\dot{q}_j(i) = 10^{-3}$ for each j, i index pairs. The maximum allowed number of steps in the optimization was 300. The Gradient Descent algorithm was realized as $\dot{q}(next) = \dot{q}(now) - \alpha_2 \nabla\Phi(now)$ with $\alpha_2 \in [10^{-6}, 10^{-4}]$. To achieve fast convergence, the actual value of α_2 was increased by the factor 1.2 for the next cycle if $\|\nabla\Phi\|$ was reduced in the previous step, but it was decreased by dividing it with 1.2 if this value have increased. However, its value was kept between the above given range. In the first series a relatively fast trajectory was tracked by using the above parameters. In a basic settings $\Delta x = 10^{-3}\,[rad]$ or $[m]$, $P_X = 3$, $P_{\dot{q}_{out}} = P_{\dot{q}_{in}} = 4$, $\Delta\dot{q} = 0.8\,[rad \cdot s^{-1}]$ or $[m \cdot s^{-1}]$ were chosen, and various trajectories were considered with its use. Figures. 3.29 and 3.30 reveal that in this manner it was possible to track a relatively fast motion (in Fig. 3.31 the joint coordinate values are in the range of approximately ±10 $[rad \cdot s^{-1}]$ or $[m \cdot s^{-1}]$).

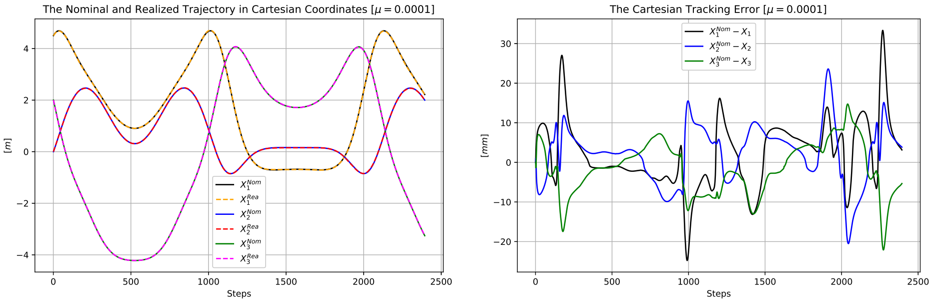

Figure 3.29: The trajectory tracking (LHS) and its error (RHS) in Cartesian Coordinates for fast motion

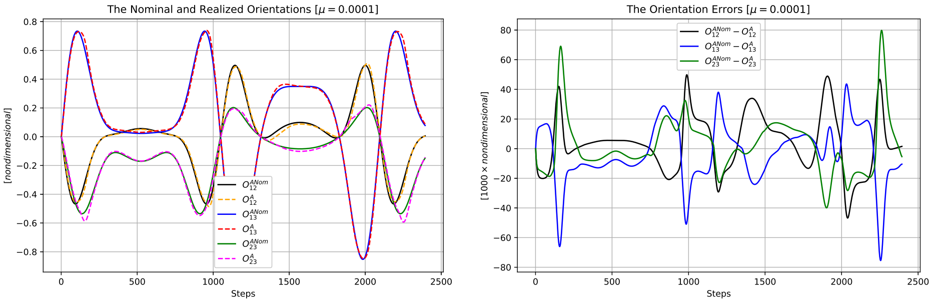

Figure 3.30: The orientation tracking (LHS) and its error (RHS) for fast motion

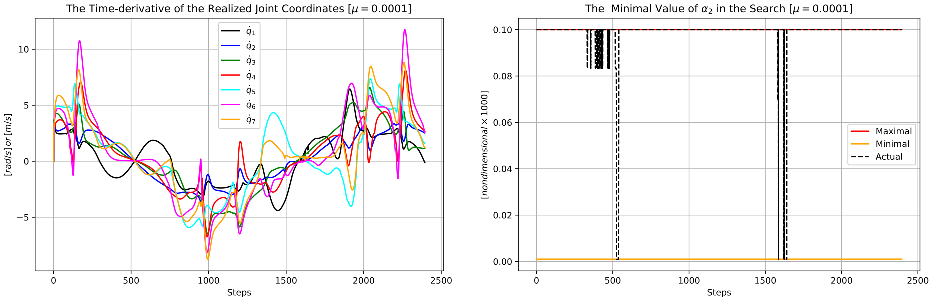

Figure 3.31: The $\dot{q}$ values (LHS) and the variation of the adaptive speed parameter α_2 (RHS) for fast motion

Figure. 3.31 testifies that the control feedback caused quite considerable fluctuation in the joint coordinate time-derivatives that seem to be superposed over a massive trend that can be observed in the figure of the realized joint coordinates in Fig. 3.32. The ambiguity of the solution is evident.

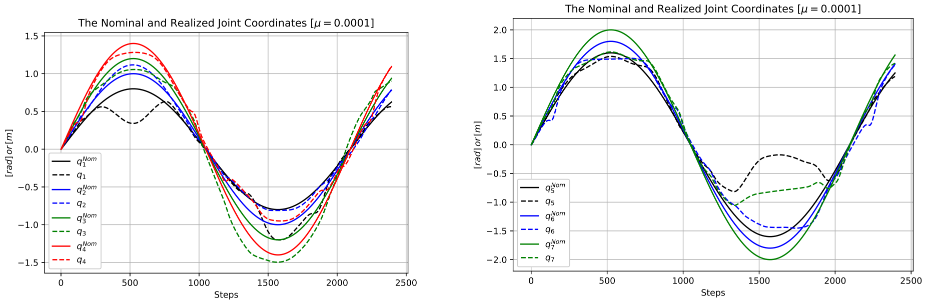

Figure 3.32: The realized q values and the joint coordinates that were used for bringing about the realizable x^N trajectory for fast motion

In the next step tracking a slower motion (in Fig. 3.35 the joint coordinate values are in the range of approximately ± 2 $[rad \cdot s^{-1}]$ or $[m \cdot s^{-1}]$) was considered. The results are given in Figs. 3.33, 3.34, and 3.35 that reveal that the joint coordinate time-derivative limitations had less significant effects and more precise tracking was possible.

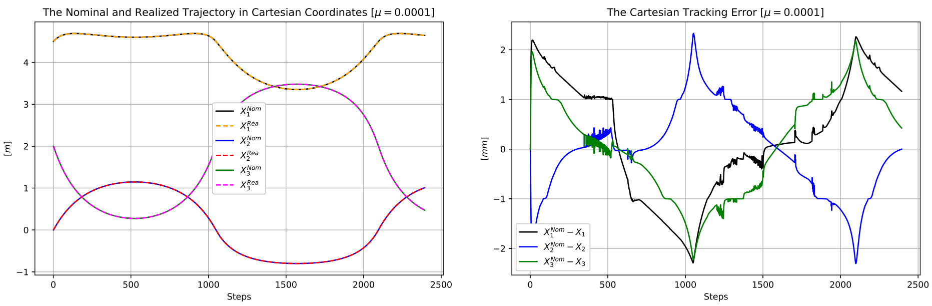

Figure 3.33: The trajectory tracking (LHS) and its error (RHS) in Cartesian Coordinates for slower motion

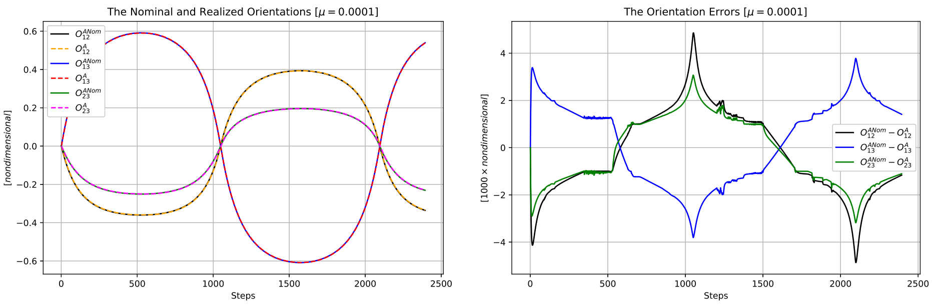

Figure 3.34: The orientation tracking (LHS) and its error (RHS) for slower motion

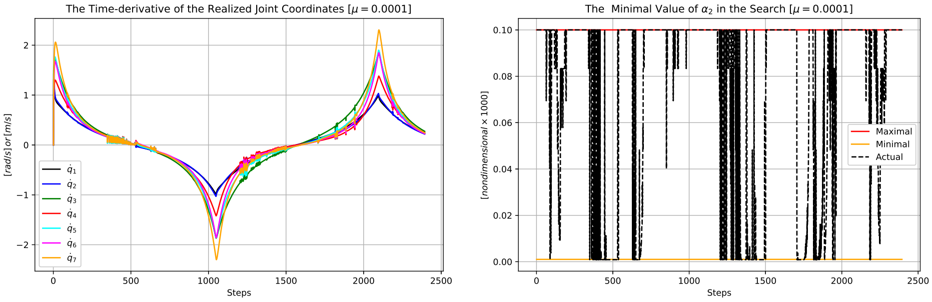

Figure 3.35: The $\dot{q}$ values (LHS) and the variation of the adaptive speed parameter α_2 (RHS) for slower motion

For a stopping motion (i.e., for motions that asymptotically approach a fixed position and orientation) simulations are presented in Figs. 3.36, and 3.37.

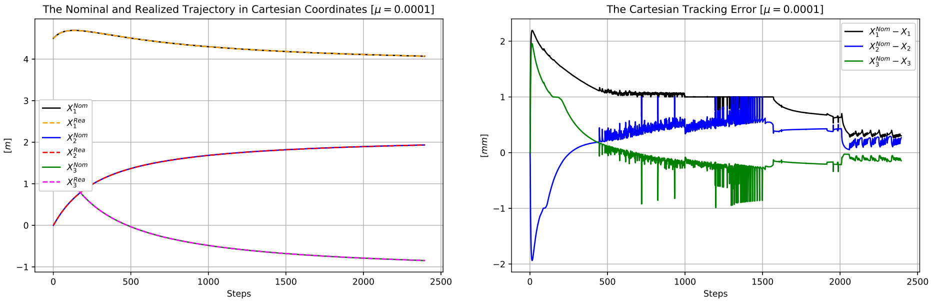

Figure 3.36: The trajectory tracking (LHS) and its error (RHS) in Cartesian Coordinates for stopping motion

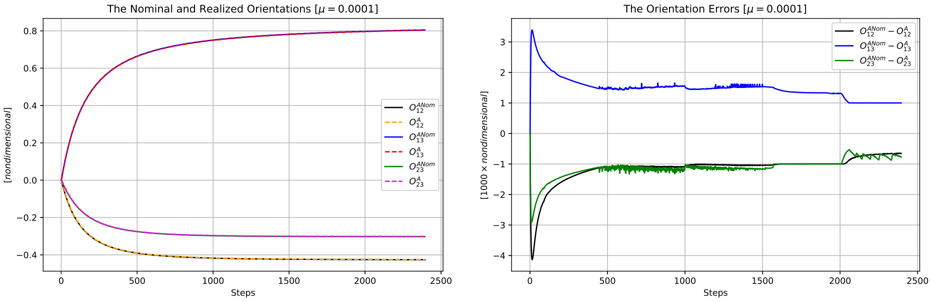

Figure 3.37: The orientation tracking (LHS) and its error (RHS) for stopping motion

To illustrate the ability of this formalism for relaxing the trajectory tracking precision requirements Δx was increased from 10^{-3} to $10^{-2}\,[rad]$ or $[m]$, $P_X = 3$, and $\Delta\dot{q}$ was decreased form 0.8 to $0.4\,[rad \cdot s^{-1}]$ or $[m \cdot s^{-1}]$ that has the following qualitative meaning: greater tracking errors are allowed with smaller penalty (cost) contributions, in the same tracking of too fast trajectory part was hindered by increasing the penalty value even for smaller $\dot{q}$ components. In harmony with the expectations, Fig. 3.38 displays the corruption of the trajectory and orientation tracking. On the basis of Fig. 3.39 it can be stated that the ambiguity of the solution was so utilized that at the cost of a considerable decrease in the $|\dot{q}_4|$, $|\dot{q}_5|$, $|\dot{q}_6|$ and $|\dot{q}_7|$ the $|\dot{q}_1|$, $|\dot{q}_2|$ and $|\dot{q}_3|$ components were increased.

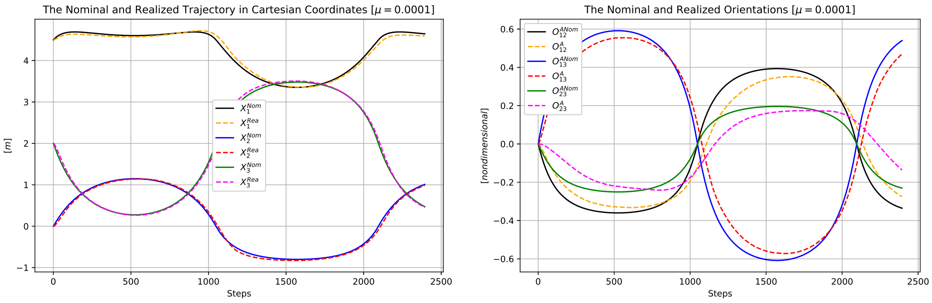

Figure 3.38: The trajectory tracking (LHS) and orientation tracking (RHS) in Cartesian Coordinates for more relaxed slower motion

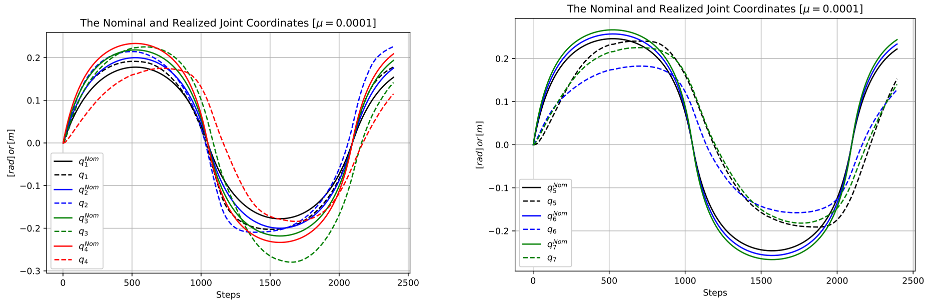

Figure 3.39: The realized q values and the joint coordinates that were used for bringing about the realizable x^N trajectory for the more relaxed slower motion

The above figures well illustrate the ability of the suggested solution to play with ambiguity of the solution of the redundant problem. The nominal trajectory to be tracked were brought about by the nominal joint coordinates $\{q_i^N(t)\}$ that were not found to be optimal since the realized $\{q_(t)\}$ solutions that produced good tracking for the position and the orientation of the end effector are considerably different to each other. The sensitivity of $\dot{q}_i(t)$ in participation of the generated motion of the end effector depend on the actual orientation of the robot arm. Since the optimization generally tries to provide the solution with the most sensitive joint coordinates, their role can change in time. Figures. 3.29 and 3.33 reveal that the fast motion swept a wider range of displacement than the slower one. Therefore different ranges of the configuration of the arm was swept by them. Figures. 3.31 and 3.35 reveal that in the case of the fast motion the joint coordinate velocities quite frequently changed role than in the case of the slow motion. In Fig. 3.35 the joint coordinates of higher indices took rather part in the realization of the motion: for shorter effective arm segment faster motion is necessary in the case of revolute joints. (More precise interpretation of the observations cannot be given for the nontrivial results of a strongly nonlinear task.) As a consequence of the integration of the joint velocities the participation of the different joint coordinates in the realized motion (Figs. 3.32 and 3.39) considerably differs from the nominal variation of the joint coordinates that generated the motion of the end effector.

3.5 Tackling Actuator Saturation in Fixed Point Iteration-based Adaptive Control

The limited output of various drives means a challenge in controller design whenever the acceleration need of the nominal trajectory to be tracked temporarily exceeds the abilities of the saturated control system. The prevailing control design methods can tackle this problem either in a single theoretical step or in two consecutive steps. In this latter case in the first step the design happens without taking into account the actuator constraints, then apply a saturation compensator if the phenomenon of windup is observed (e.g., [101–103]). In the Fixed Point Iteration-based Adaptive Control (FPIAC) that has been developed as an alternative of the Lyapunov function-based approach the actuator saturation causes problems in its both elementary levels: in the kinematic/kinetic level where the desired acceleration is calculated, and in the iterative process that compensates the effects of modeling errors of the dynamic system under control and that of the external disturbances.

In the case of the adaptive control the precision further can be improved by switching on the adaptive loop. However, if the control drive is not able to exert enough force, the adaptive law may produce enormous deformation because the realized acceleration can never achieve the desired one. To evade this situation the following steps can be done within the frames of the FPIAC scheme:

1. Stop the increase/decrease of $e_{int}(t)$ over/under a limited value $e_{int_{max}}/{-e_{int_{max}}}$;
2. Apply an interpolation parameter $\lambda \in [0,1]$ and use the weighted desired 2nd time-derivative $\ddot{q}^{Des} = \ddot{q}^{Des}_{PID}\lambda + \ddot{q}^{Des}_{PD}(1-\lambda)$;
3. Introduce a maximal absolute value for the *deformed value* with a sigmoid function $\overline{\ddot{q}^{Def}} = \ddot{q}_{max}\sigma\left(\ddot{q}^{Def}/\ddot{q}_{max}\right)$, and use this limited $\overline{\ddot{q}^{Def}}$ value as the input of the approximate dynamic model;
4. Introduce a maximal absolute value for the torque and apply it at the output of the approximate dynamic model Q as $Q_{lim} = Q_{max}\sigma\left(Q/Q_{max}\right)$, and exert this limited force to the controlled system;
5. Set the weighting parameter as $\lambda = 1 - \sigma(|Q_{lim}|/Q_{max})$.

The above rules guarantee that the integrated error term cannot diverge, the adaptive deformation cannot not diverge, and the controller's drive is not overloaded. As a consequence, within the range of the abilities of the drive precise trajectory tracking can be achieved, while outside of this region the tracking precision is corrupted but the control signal remains stable. The operation of this modified adaptive controller will be investigated in the sequel via simulations.

3.5.1 Simulation Results

To exemplify the operation of the method nominal trajectories with small (i.e., 0.25) and big (i.e., 1.0) amplitude were considered for the modified van der Pol oscillator [104], in which the modification consisted in adding a nonlinear damping term that is proportional to the square of the velocity of the moving body that is typical in the case of motion in a turbulent fluid. The equation of motion of the system is given in (3.38) in which Q denotes the control force. This system has an unstable equilibrium state in $q = 0, \dot{q} = 0$, and it is excited if $q^2 < d$, and damped if $q^2 > d$.

$$m\ddot{q} = -kq - b(q^2 - d)\dot{q} - c\,\mathrm{sign}(\dot{q})\dot{q}^2 + Q \quad , \tag{3.38}$$

The approximate model in (3.39) was a simple damped linear oscillator with the parameters $m_a = 6.0\,[kg]$, $k_a = 200.0\,[N \cdot m]$, and $b_a = 15.0\,[N \cdot s \cdot m^{-1}]$. It had a stable equilibrium in the same state.

$$m_a\ddot{q} = -k_a q - b_a\dot{q} + Q \tag{3.39}$$

The control parameters were: $\Lambda = 2.5\,[s^{-1}]$, $K_c = 2 \times 10^3\,[m \cdot s^{-2}]$, $B_c = -1$, and $A_c = 0.125/K_c$. In (3.14) $\sigma_1(x) \equiv \tanh(x)$, in the limitations $\sigma(x) \equiv x/(1+|x|)$ was used. The exact model parameters are given in Table 3.7

Table 3.7: Parameters of the controlled system

Parameter	**Exact Value**
Mass	$m = 20.0\,[kg]$
Spring's stiffness	$k = 100.0\,[N \cdot m^{-1}]$
Damping coeff.	$b = 3.0\,[N \cdot s \cdot m^{-3}]$
Turbulent friction coeff.	$c = 6.0\,[N \cdot s^2 \cdot m^{-2}]$
Excitation/Damping separator	$d = 0.5\,[m^2]$

Figures. 3.40–3.43 reveal that without force limitations the adaptive controller efficiently can track the prescribed motion and well compensates the effects of the modeling errors in the PID-type control.

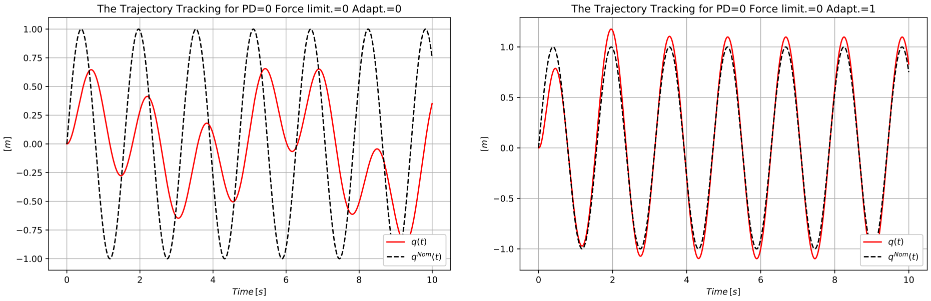

Figure 3.40: Non-adaptive (LHS) and adaptive (RHS) PID-type trajectory tracking without force limitation

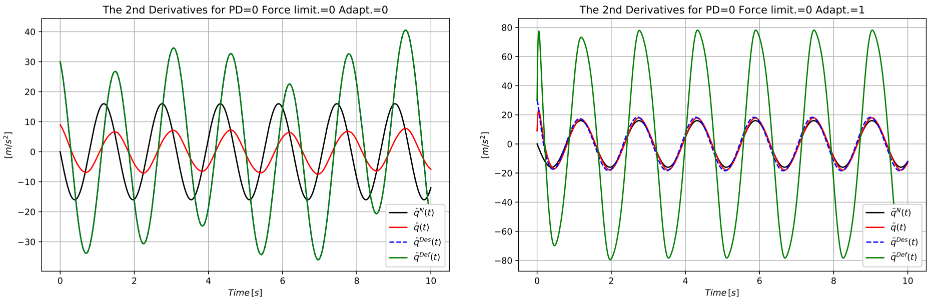

Figure 3.41: Non-adaptive (LHS) and adaptive (RHS) $\ddot{q}$ values without force limitation in a PID-type control

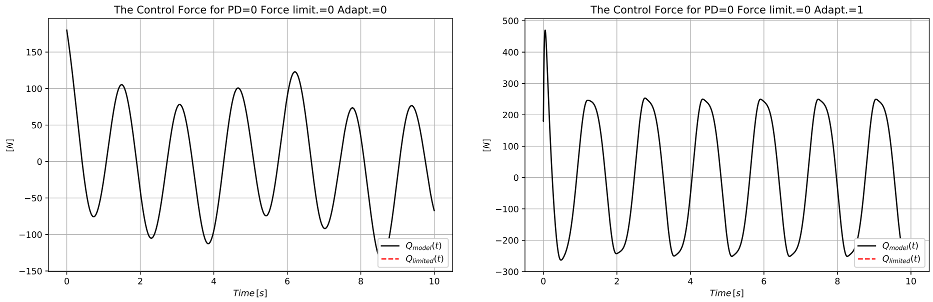

Figure 3.42: Non-adaptive (LHS) and adaptive (RHS) control force without force limitation in a PID-type control

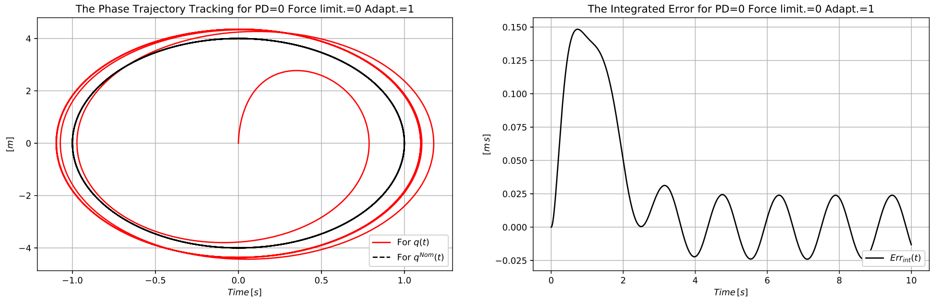

Figure 3.43: The phase trajectory tracking (LHS) and the integrated error (RHS) of the adaptive controller without force limitation in a PID-type control

The situation drastically changes if in the same system the limitation $Q_{max} = 300.0\,[N]$ is switched on. Figure. 3.44 reveals the corruption of the trajectory tracking, while in Fig. 3.45 the windup of the adaptive deformation and the calculated control force (that cannot be exerted) can be observed. Also, according to Fig. 3.46, the integrated tracking error takes great values.

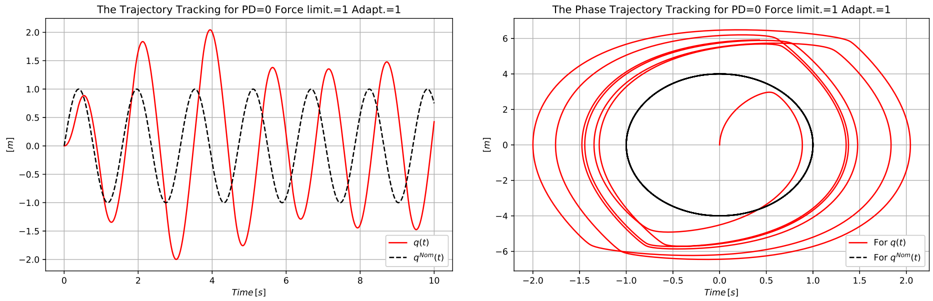

Figure 3.44: The trajectory (LHS) and phase trajectory (RHS) tracking in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$

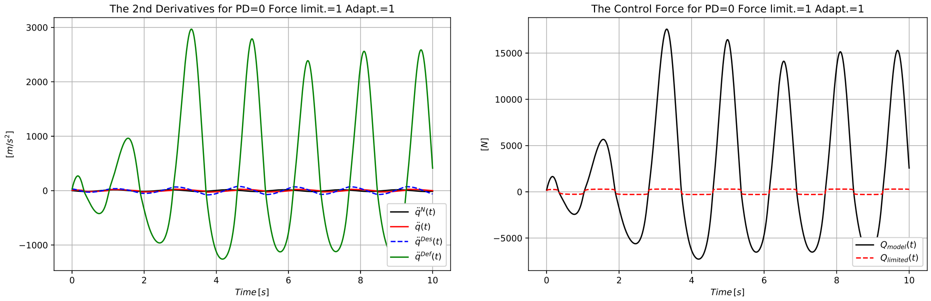

Figure 3.45: The $\ddot{q}$ values (LHS) and the control force (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$

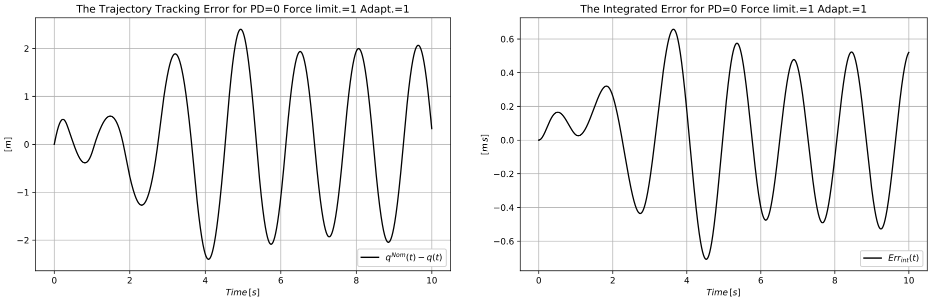

Figure 3.46: The trajectory tracking error (LHS) and the integrated error (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$

Figures. 3.47–3.49 testify that no such problem occurs if the amplitude of the nominal motion is small: in this case the saturation of the forces does not play important role in the operation of the adaptive PID controller.

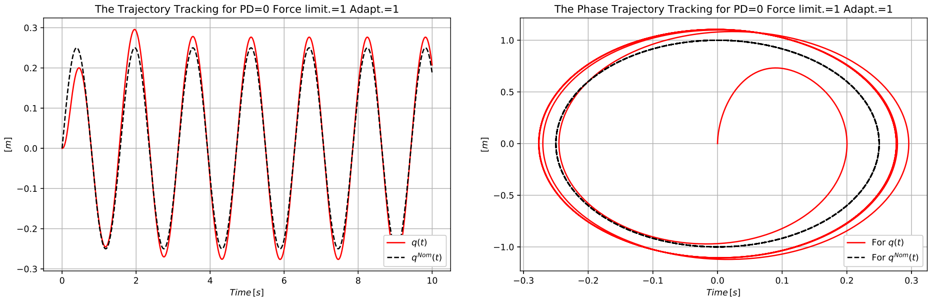

Figure 3.47: The trajectory (LHS) and phase trajectory (RHS) tracking in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

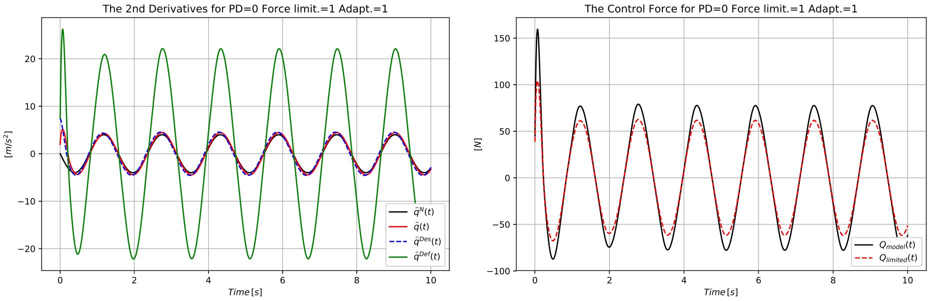

Figure 3.48: The $\ddot{q}$ values (LHS) and the control force (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

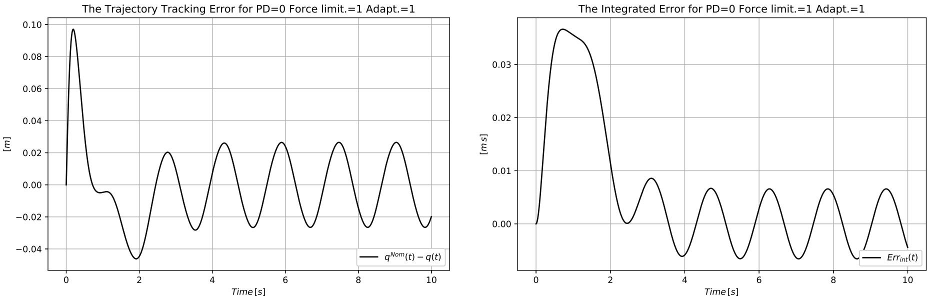

Figure 3.49: The trajectory tracking error (LHS) and the integrated error (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

To show the effects of the introduction of the limitation in the integrated error with $e_{int_{max}} = 0.05\,[m \cdot s]$, $\ddot{q}_{max} = 10^3\,[m \cdot s^{-2}]$, and the application of the interpolation parameter λ between the PID and PD-type kinematic tracking requirements the motion of big nominal amplitude was considered. It is evident that the tracking error is big (the controller's drive does not have the necessary capacity), however, the effect of windup has been evaded in spite of the saturation of the control signal. The interpolation factor $\lambda(t)$ was closer to 1 than to 0 that means that rather the PID-type tracking strategy dominated the kinematic design.

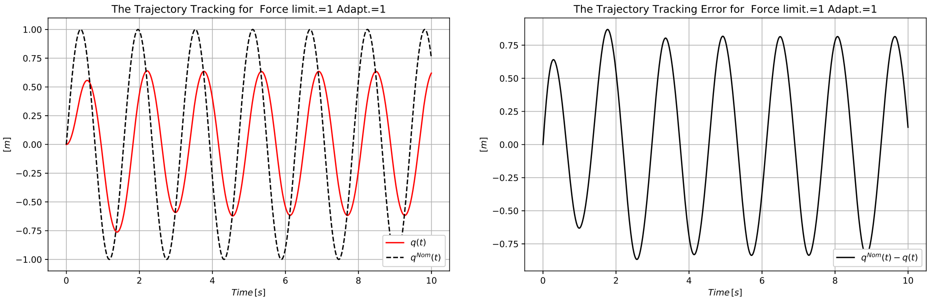

Figure 3.50: The trajectory tracking (LHS) and the tracking error (RHS) in the suggested novel adaptive control for big amplitude of the nominal motion

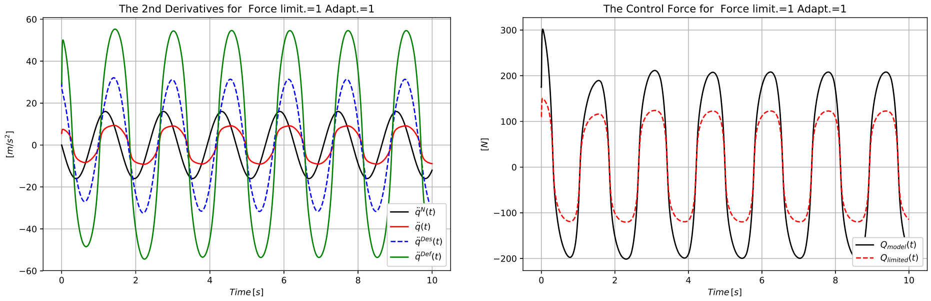

Figure 3.51: The $\ddot{q}$ values (LHS) and the control force (RHS) in the suggested novel adaptive control for big amplitude of the nominal motion

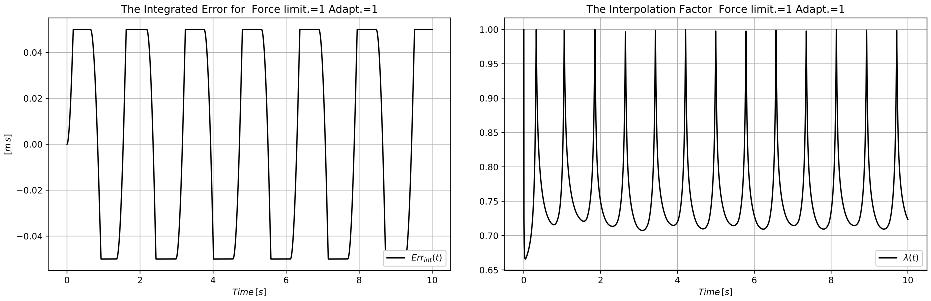

Figure 3.52: The integrated error (LHS) and the interpolation parameter λ (RHS) in the suggested novel adaptive control for big amplitude of the nominal motion

The counterparts of Figs. 3.50–3.52 (belonging to the big amplitude of nominal motion) are Figs. 3.53–3.55 for the nominal trajectory of small amplitude. According to the figures it can be stated that the saturation effects did not play significant role in this control.

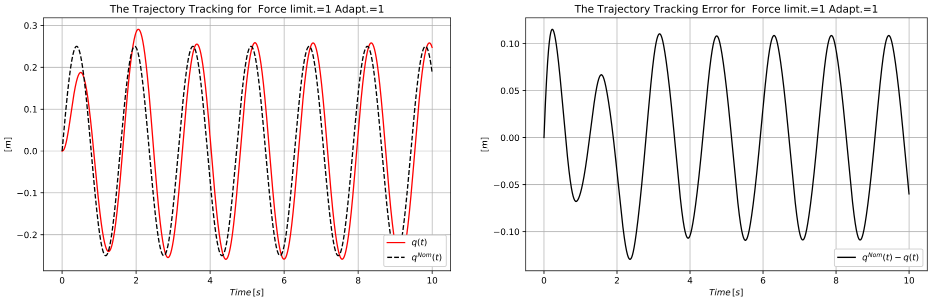

Figure 3.53: The trajectory tracking (LHS) and the tracking error (RHS) in the suggested novel adaptive control for small amplitude of the nominal motion

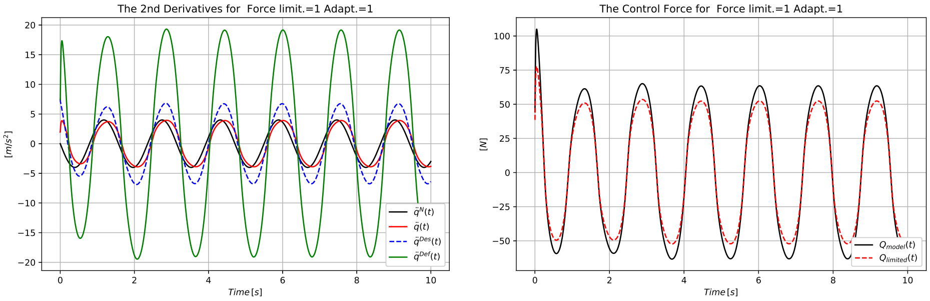

Figure 3.54: The $\ddot{q}$ values (LHS) and the control force (RHS) in the suggested novel adaptive control for small amplitude of the nominal motion

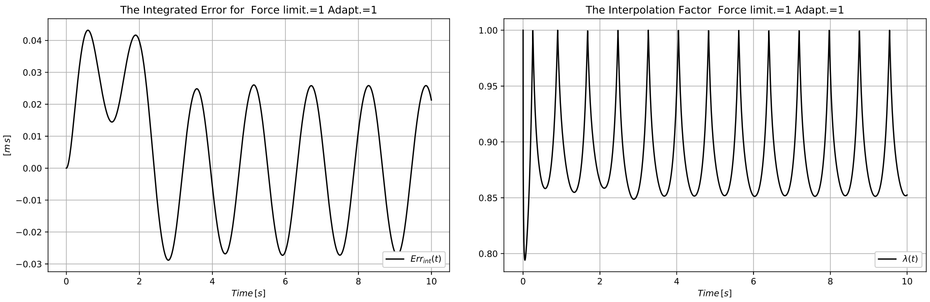

Figure 3.55: The integrated error (LHS) and the interpolation parameter λ (RHS) in the suggested novel adaptive control for small amplitude of the nominal motion

3.5.1.1 Simulations with LuGre Friction Model

In these simulations the approximate dynamic model of the van der Pol oscillator contained only a viscous friction force model. To test the abilities of the adaptive controller in the exact system an additional, LuGre-type friction model was built up. In this model the friction effects are described the use of a complementary dynamic variable $z(t)$ that cannot be directly observed by the controller. The FPI-based approach is especially convenient when some hidden, dynamically coupled subsystem disturbs the motion of the controlled physical system. The adaptive mechanism automatically makes the necessary correction without the need of developing complicated observers for the hidden state variable. The parameters of the friction model are given in Table 3.8. The control parameters were: $\Lambda = 2.5\,[s^{-1}]$, $K_c = 2\times 10^3\,[m\cdot s^{-2}]$, $B_c = -1$, and $A_c = 5.0/K_c$ in (3.14).

Table 3.8: The LuGre model parameters in (3.30) of the simulations (they were set experimentally in order to keep the friction forces in the order of magnitude of that of the simple viscous model)

Parameter	**Value**
$\sigma_0\ [Nm^{-1}]$	5000.0
$\sigma_1\ [Nsm^{-1}]$	1000.0
$F_C\ [N]$	10.0
$F_S\ [N]$	20.0
$F_v\ [Nsm^{-1}]$	10.0
$v_s\ [ms^{-1}]$	1.0

I. Simulations without force limitation

At first simulation results are presented without force limitation. Figures 3.56 and 3.58 testify that both the phase trajectory and the trajectory tracking remained precise in the adaptive control, in spite of the not modeled friction force in Fig. 3.59. It can be observed that in the control force considerable terms appeared for the compensation of the friction force (Fig. 3.57). The last figure also reveals that a great extent of adaptive deformation was necessary for the compensation of the effects of the not modeled friction force. The phase trajectory of the hidden dynamic variable of the LuGre model $z(t)$ well explains the nature of this model, too (Fig. 3.59). The phenomenon of stick-slip can be well identified in this figure.

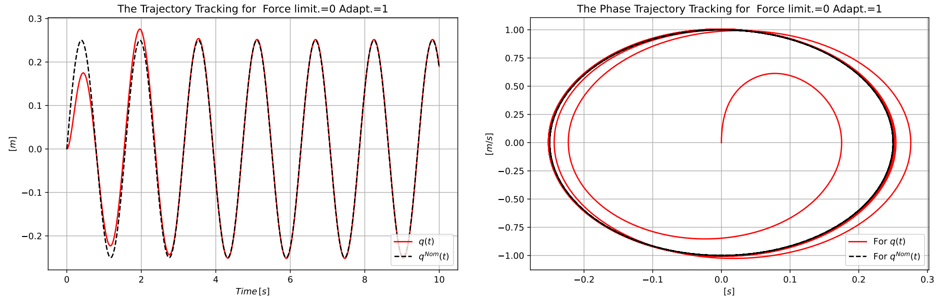

Figure 3.56: The trajectory (LHS) and phase trajectory (RHS) tracking in the adaptive PID control without force limitation and small amplitude of the nominal motion

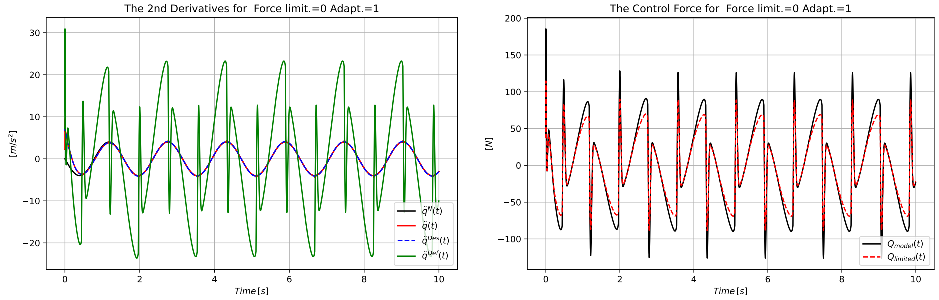

Figure 3.57: The $\ddot{q}$ values (LHS) and the control force (RHS) in the adaptive PID control without force limitation and small amplitude of the nominal motion

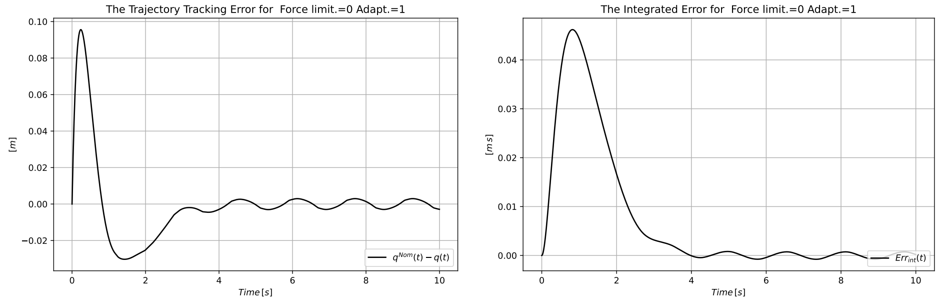

Figure 3.58: The trajectory tracking error (LHS) and the integrated error (RHS) in the adaptive PID control without force limitation and small amplitude of the nominal motion

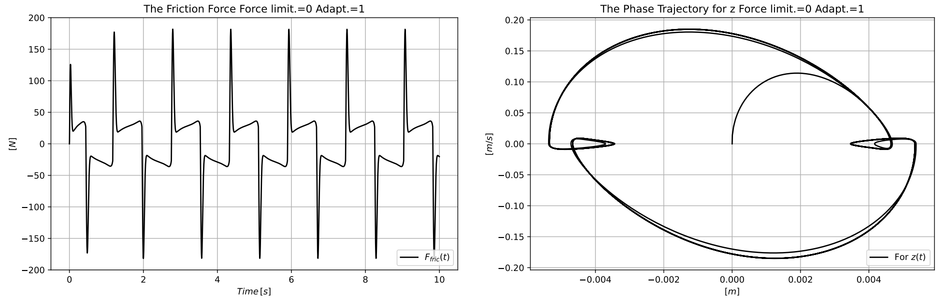

Figure 3.59: The Friction Force (LHS) and the Phase trajectory for z (RHS) in the adaptive PID control without force limitation small amplitude of the nominal motion

II. Simulations with force limitation

he simulation results displayed in Figs, 3.60, 3.61, 3.62, 3.63 have similar properties and allow similar consequences as that of the free of friction case.

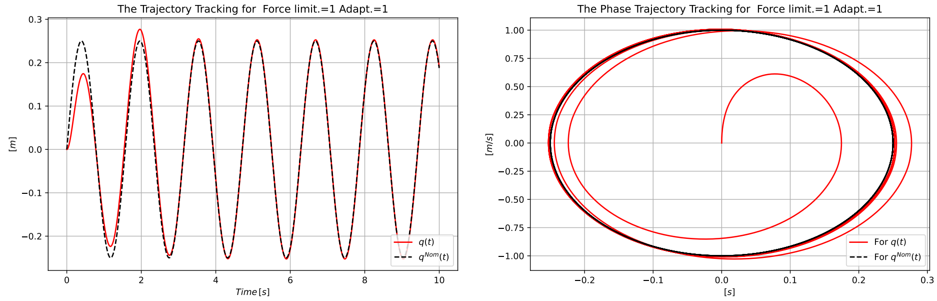

Figure 3.60: The trajectory (LHS) and phase trajectory (RHS) tracking in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

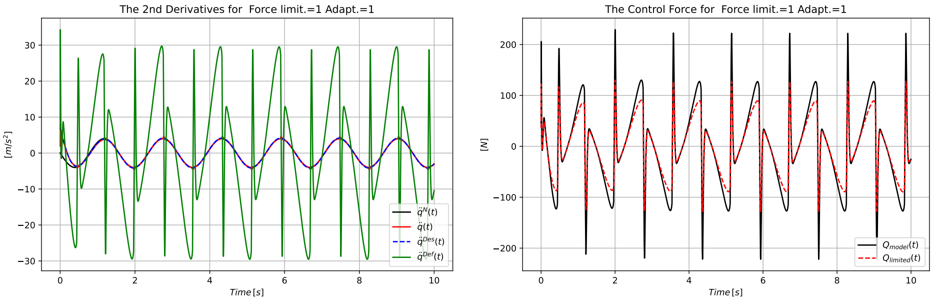

Figure 3.61: The $\ddot{q}$ values (LHS) and the control force (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

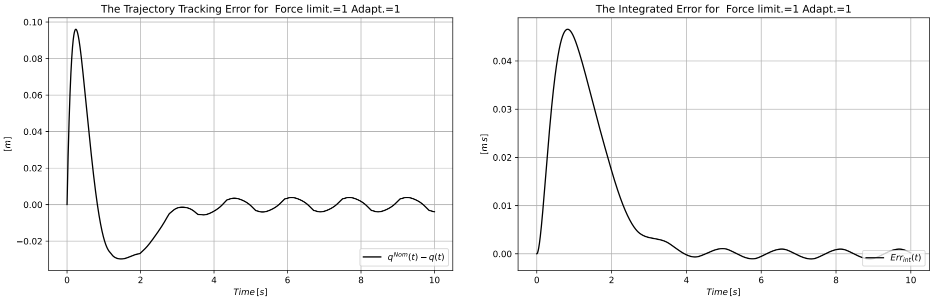

Figure 3.62: The trajectory tracking error (LHS) and the integrated error (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

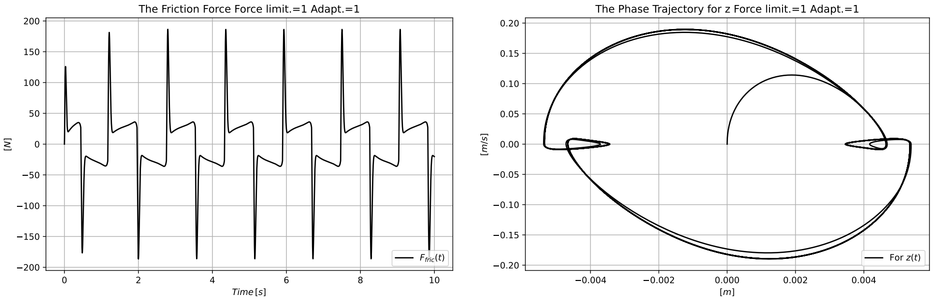

Figure 3.63: The Friction Force (LHS) and the Phase trajectory for z (RHS) in the adaptive PID control in the case of force limitation $Q_{max} = 300.0\,[N]$ and small amplitude of the nominal motion

3.6 Thesis Statement II

I have elaborated further modification of the Receding Horizon Controller. By directly incorporating the dynamic model into the cost function calculated over a horizon, I have completely eliminated the use of the constraint term from the formalism. I have

shown that by the application of a transition between the simple Gradient Descent and the fast Newton-Raphson Algorithms an efficient method can be developed for finding the local minima. In this manner strong penalization of high forces became possible, however, the method did not guarantee the evasion of actuator saturation and windup problems in the Fixed Point Iteration-based adaptive control.

To tackle actuator saturation and windup problems I have elaborated a complementary method in which a hypothetical heavy device model was applied in the optimization of the trajectory, and the so obtained optimized trajectory was adaptively tracked by a realistic not heavy approximate dynamic model. The main benefit of this approach is that the optimization's mathematical structure can be completely separated from that of the adaptive tracking.

Related own publications: **[A. 3], [A. 4], [A. 5] and [A. 6]**.

Chapter 4

Investigation of the Cooperation of Noise Filtering Methods With Fixed Point Iteration-based Adaptive Techniques

In Chapter 1 I briefly mentioned that in dynamic control problems the presence of observation or sensor noise is a general problem. The special structure of the Fixed Point Iteration-based approach makes one expect that the noise issues have enhanced significance in this case. The success of the Acceleration Feedback Controllers (e.g., [105]) that have to cope with similar problems also confirm the idea that useful investigations can be done in this direction. Fixed point iteration-based control specific preliminaries as ad hoc ideas were already published in [67–70]. In Subsection 3.3.2 I already applied the traditional low pass filter technique that was borrowed from [66, 99]. In this Chapter I summarize the results of my own novel investigations.

The first part investigates a very drastic noise filtering technique that was introduced to support the operation of the adaptive control. Its basic idea is affine approximation of the various derivatives within successive moving windows. This idea was checked in cooperation of a special continuous variant of fixed point transformations. The investigations were made for the modified van der Pol oscillator that had an additional quadratic drag force term also used in Subsection 3.5.

The second part outlines the comparison between Unscented Kalman Filter (UKF) and two methods based on Fixed Point Operation-based adaptive controllers.

4.1 Using High Frequency Noisy Signal Filtering

In this approach, the noisy signal of the direct observation modeled with a noise term $\mathcal{N}(t)$ added to the exact value $q(t)$ as $q_o(t) = q(t) + \mathcal{N}(t)$ was tracked by the filtered (smoothed) signal $q_s(t)$ according to the differential equation and initial conditions given in (4.1)

$$\begin{aligned} &\left(\Lambda_{filt} + \frac{\mathrm{d}}{\mathrm{d}t}\right)^3 q_s(t) = \Lambda_{filt}^3 q_o(t) \\ &q_s(t_0) = 0\,, \dot{q}_s(t_0) = 0\,, \ddot{q}_s(t_0) = 0\,, \end{aligned} \tag{4.1}$$

with $0 < \Lambda_{filt} =$ constant. This idea has been borrowed from [66, 99]. In the frequency domain at zero frequency it has the transfer function 1, while for high frequencies (very drastic rejection) it is $\propto s^{-3}$ (s is the variable of the Laplace transform), that is it realizes drastic suppression for the high frequency components. This method was applied for smoothing the optimized trajectory in subsection 3.3.2 that was adaptively tracked by the use of an available approximate dynamic model of the actual system under control.

4.2 Application of Simple Moving Window with Affine Signal Approximation

For noise filtering the following idea was used: the exact values q_e, $\dot{q}_e$, $\ddot{q}_e$ were obtained from the simulations via Euler integration and use of the exact model parameters and state variables. The observed coordinate value was burdened with some measurement noise as $q_o(t) = q_e(t) + N_A[\text{rand}()]$. Instead of the noisy observed $q_o(t)$ values the controller used the smoothed $q_s(t)$ values that were calculated by fitting an affine form to the last L discrete measured values. A buffer of length $L \in \mathbb{N}$ was introduced as a moving signal sampling window that was filled in with the latest observed values as $\{q_o(t-L+0), q_o(t-L+1), \dots q_o(t-L+L)\}$, i.e., $\{q_o(t-L+m)\}$, $m = 0, 1, \dots, L$, and these values were approximated as $\{a_0 m + b_0\}$, i.e., for obtaining the first derivative an *affine form of the signal* was considered. Via fitting the parameters a_0, b_0 with the least squares error, the smoothed value $q_s(t) = a_0 L + b_0$ was chosen at the end of the window.

In the place of the direct numerical derivative of the noisy signal the filtered approximation (i.e., the observed values) $\dot{q}_o(t) = \frac{q_s - q_s(t-1)}{\delta t}$ was written. Following that, the above approximation for the $\dot{q}_o(t)$ values were put into the grid points $\{\dot{q}_o(t-L+0), \dot{q}_o(t-L+1), \dots \dot{q}_o(t_{i-L+L})\}$, and were approximated with the same affine form as $\{a_1 m + b_1\}$ in quite similar manner providing $\dot{q}_s = a_1 L + b_1$ to provide the filtered derivatives as $\ddot{q}_s$. This scheme can be continued for obtaining the higher order derivatives.

The integrated error e_{int} was computed by the use of $q_s(t)$. Evidently, the significance of the noise amplitude N_A in a control task depends on the dynamics of the nominal trajectory to be tracked.

4.3 Continuous Time Fixed Point Iteration-based Control

In the following, another form of Adaptive Deformation block (from Fig. 3.1) will realize the following sequence of the deformed signals ([106]):

$$\ddot{q}^{Def}(0) = \ddot{q}^{Des}(0) \ , \tag{4.2a}$$

$$\ddot{q}^{Def}(i+1) = \ddot{q}^{Def}(i) + A\sigma\left(\frac{f\left(\ddot{q}^{Def}(i)\right) - \ddot{q}^{Des}(i+1)}{w}\right) \ , \tag{4.2b}$$

in which $\sigma(x)$ is a monotonic increasing sigmoid function with the properties $\sigma(-\infty) = -1$, $\sigma(\infty) = 1$, $\sigma(0) = 0$, and $A, w \in \mathbb{R}$ are adaptive control parameters. If $\ddot{q}^{Des}(t)$ only slowly varies, i.e., for the above iteration it approximately can be considered as a constant, in the vicinity of the solution of the control task $\ddot{q}_\star$ for which $f(\ddot{q}_\star) = \ddot{q}^{Des}$ first order Taylor series approximation can be done for the iteration:

$$f(\ddot{q}) = f(\ddot{q} - \ddot{q}_\star + \ddot{q}_\star) \approx f(\ddot{q}_\star) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\ddot{q}_\star} (\ddot{q} - \ddot{q}_\star) \ , \tag{4.3a}$$

$$\ddot{q}(i+1) = \ddot{q}(i) + \frac{A}{w} \left.\frac{\mathrm{d}\sigma}{\mathrm{d}x}\right|_0 \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\ddot{q}_\star} (\ddot{q}(i) - \ddot{q}_\star) \ , \tag{4.3b}$$

$$\ddot{q}(i+1) - \ddot{q}_\star \approx \left[1 + \frac{A}{w} \left.\frac{\mathrm{d}\sigma}{\mathrm{d}x}\right|_0 \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{\ddot{q}_\star}\right] (\ddot{q}(i) - \ddot{q}_\star) \ . \tag{4.3c}$$

Evidently, $\ddot{q}_\star$ is the fixed point of the deformation function, and if for the quantity in (4.3c) it holds that $-1 < [\cdot] < 1$, the iteration converges to its fixed point. With regard to convergence issues relevant details were written in Subsection 3.1.3

Following the idea of the paper [106], for a short cycle time τ and a slowly varying $\ddot{q}^{Des}$ value in (4.2b) can be interpreted as follows:

$$\begin{aligned} &\frac{\mathrm{d}\ddot{q}^{Def}}{\mathrm{d}t} \approx \frac{\ddot{q}^{Def}(i+1) - \ddot{q}^{Def}(i)}{\tau} = \\ &= \frac{A}{\tau}\sigma\left(\frac{f\left(\ddot{q}^{Def}(i)\right) - \ddot{q}^{Des}}{w}\right) \ , \end{aligned} \tag{4.4}$$

that indicates that instead making iterations between the consecutive digital control cycles finer approach can be achieved by continuously tuning the variable $\ddot{q}^{Def}$. Something similar happens in the application of the classical Luenberger observer in the control of Linear Time-invariant (LTI) systems [107].

For designing a Luenberger observer, where the exact state variable $q(t)$ is not precisely known, because the measurable value $q^{Meas}(t)$ is burdened with some unknown additional stochastic measurement noise $\mathcal{N}(t)$ as $q^{Meas}(t) = q(t) + \mathcal{N}(t)$, the motion of an estimated value $\hat{q}(t)$ is computed and used for the calculations that tracks the noisy signal via its first time-derivative, starting from some plausible initial estimate $\hat{q}(t_0)$ in (4.5a). This equation of motion is similar to that of $q(t)$ in (4.5b), but it also contains an additional drift term that pulls $\hat{q}(t)$ toward $q^{Meas}(t)$ through $\dot{\hat{q}}(t)$. The application of the first derivative corresponds to a gentle low pass type noise filtering. From (4.5c) it is clear that if the noise is not too big, i.e., $q^{Meas}(t) \approxeq q(t)$, for a stable system matrix A an appropriate additional term can speed up the convergence $\hat{q}(t) \to q(t)$. When the tracking error became small enough, for the estimation of the joint coordinate derivative the $\dot{q}(t) \approxeq \dot{\hat{q}}(t)$ can be applied independently of the calculation of the control force $u(t)$. In (4.4) some similar tracking property is formulated for $\frac{\mathrm{d}\ddot{q}^{Def}}{\mathrm{d}t}$ with limited amplitude so that $\ddot{q}^{Def}(t)$ itself inherits some noise in the FPI-based control. For dealing with nonlinear system, the limitation of the possible value by the function σ is practically desirable. The Luenberger observer works as follows:

$$\dot{\hat{q}}(t) = A\hat{q}(t) + Bu(t) + L(q^{Meas}(t) - \hat{q}), \tag{4.5a}$$

$$\dot{q}(t) = Aq(t) + Bu(t) \ . \tag{4.5b}$$

$$\text{By subtracting} \Rightarrow \frac{d}{dt}(q(t) - \hat{q}(t)) = A(q(t) - \hat{q}(t)) + L(q^{Meas}(t) - \hat{q}(t)) \ . \tag{4.5c}$$

in which $0 < L$ determines the speed of the additional drift term.

The saturation of the sigmoid function limits the speed of this drift in our case when the deformed value is far from the solution, and w determines the order of magnitude of the response error over which the maximum speed has to be applied. In the vicinity of the solution the Taylor series approximation yields

$$\frac{\mathrm{d}\ddot{q}^{Def}}{\mathrm{d}t} \approx \frac{A}{\tau w} \left.\frac{\partial f}{\partial x}\right|_{\ddot{q}_\star} \left(\ddot{q}^{Def} - \ddot{q}_\star\right) \ . \tag{4.6}$$

If the drift of $\ddot{q}_\star$ is small in comparison with that of $\ddot{q}^{Def}$ the above approximation can be further interpreted as

$$\frac{\mathrm{d}\left(\ddot{q}^{Def} - \ddot{q}_\star\right)}{\mathrm{d}t} \approx \frac{A}{\tau w} \left.\frac{\partial f}{\partial x}\right|_{\ddot{q}_\star} \left(\ddot{q}^{Def} - \ddot{q}_\star\right) \ , \tag{4.7}$$

for which, on the basis of the Jordan canonical form [108, 109], the satisfactory condition of the convergence is that the real part of each eigenvector of the matrix $\frac{A}{\tau w} \left.\frac{\partial f}{\partial x}\right|_{\ddot{q}_\star}$ must be negative.

4.4 Performance of Fixed Point Iteration-based Adaptive Control in Noise Filtering

The simulations were made for the van der Pol oscillator with additional quadratic drag force term in the equation of motion (3.38). The model data and control parameters are given in Table 4.1. The discrete time resolution of the Euler integration was $\delta t = 10^{-3}\,[s]$. The observed coordinate value was burdened with measurement noise $q_o(t) = q_e(t) + N_A[0.5 - \mathrm{rand}()]$ in which the random function generator provided values with even probability in the range $[0, 1]$, and the Noise Amplitude was $N_A = 6 \times 10^{-3}\,[m]$.

Qualitatively it can be expected that if fast motion has to be controlled, in which during the cycle time τ, and especially during the filtering time $L \cdot \tau$ the variation of the generalized coordinates is considerable, and the great velocities imply strong nonlinear control forces, the relative significance of the measuring noise of the generalized coordinates is smaller and the acceleration feedback in the adaptive approach can improve the control quality. However, for slow motion it can be expected that the acceleration feedback may corrupt the control quality and the traditional PID-type approaches may produce superior results.

Table 4.1: The Model and Control Parameters of the Simulations

Measurement	**Parameter**		
Unit	***Description***	***Exact Value***	***Approx. Value***
$[kg]$	Mass m	20.0	10.0
$[N\cdot m^{-1}]$	Stiffness k	100.0	400.0
$[N\cdot s\cdot m^{-1}]$	Damping b	3.0	2.1
$[N\cdot m^{-3}\cdot s]$	Excitation d	2.5	2.0
$[m]$	Separator a	1.5	1.2
$[N\cdot m^{-2}\cdot s^{2}]$	Drag Coeff. c	6.0	3.0
$[s^{-1}]$	Λ in (3.3)	6.0	$--$
$[s]$	τ in (4.4)	10^{-3}	$--$
$[m\cdot s^{-2}]$	w in (4.4)	6.0	$--$
$[m\cdot s^{-1}]$	A in (4.4)	-1.0	$--$

The table also contains the control parameters applied for a PID-type trajectory tracking and that of the Continuous Fixed Point Transformation in (4.4). Typical result is obtained for the trajectory and phase trajectory tracking of fast motion in Figs. 4.1 and 4.2. It is evident that in this case the adaptivity well improved the control quality in the noisy case, too.

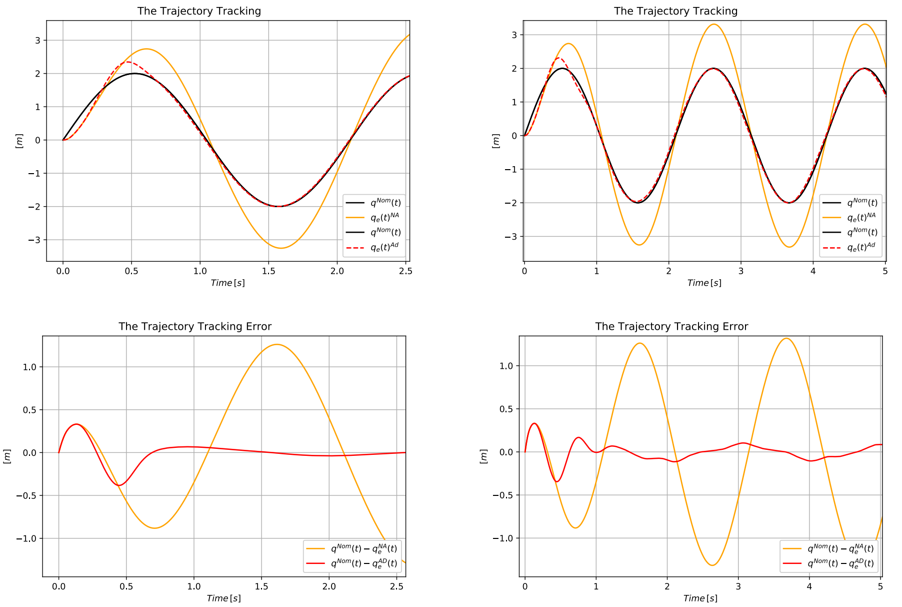

Figure 4.1: Comparison of effect of adaptivity on the: (At top: trajectory tracking) , (At bottom: related trajectory tracking error) in the noise-free [LHS] and noisy [RHS] cases of fast motion (NA: non adaptive, AD: Adaptive)

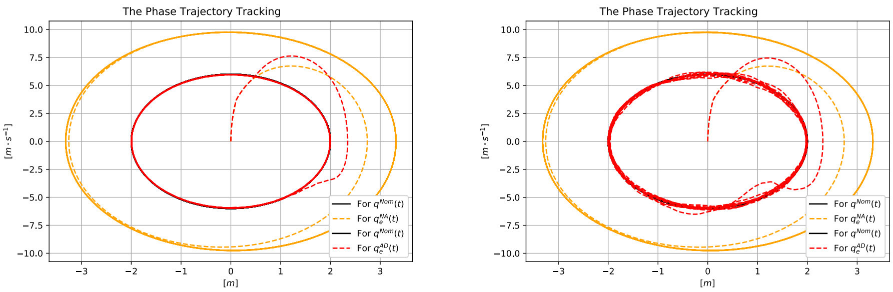

Figure 4.2: Comparison of effect of adaptivity on the phase trajectory tracking in the noise-free [LHS] and noisy [RHS] cases of "fast" motion (NA: non adaptive, AD: Adaptive)

Figure. 4.3 reveals that the noises essentially did not blocked the adaptive algorithm: the desired and the realized values were very close to each other.

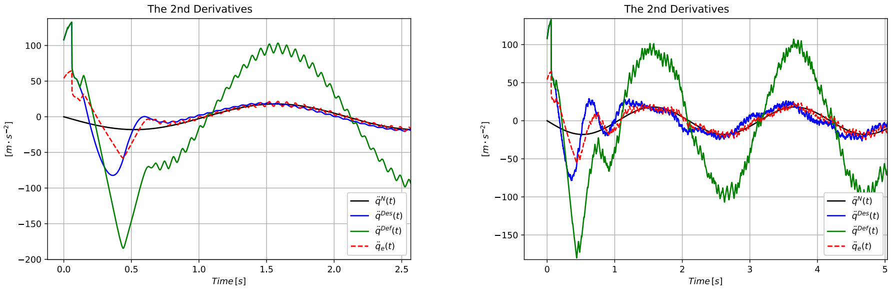

Figure 4.3: Comparison of the development of the adaptive control signal in the noise-free [LHS] and noisy [RHS] cases of fast motion

Similar observations can be done for a considerably slower motion at which the increased effect of the measurement noises can be best studied in the graphs of the phase trajectories (Figs. 4.4–4.6).

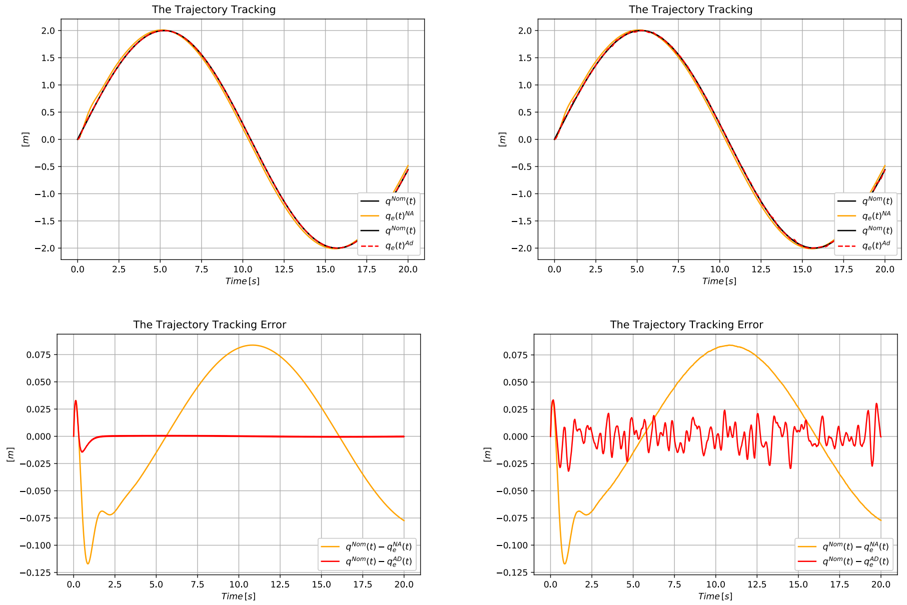

Figure 4.4: Comparison of effect of adaptivity on the: (At top: trajectory tracking) , (At bottom: related trajectory tracking error) in the noise-free [LHS] and noisy [RHS] cases of slow motion (NA: non adaptive, Ad: Adaptive)

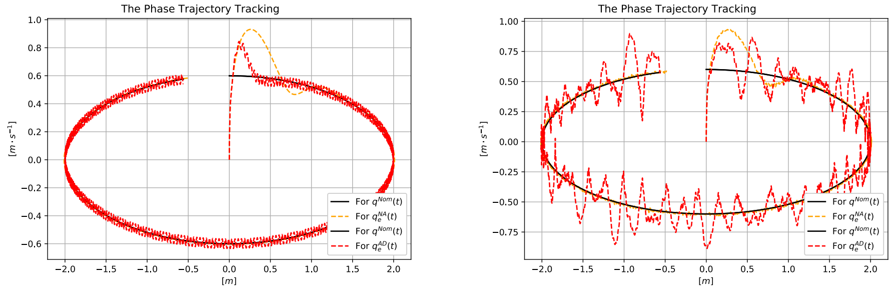

Figure 4.5: Comparison of effect of adaptivity on the phase trajectory tracking in the noise-free [LHS] and noisy [RHS] cases of "slow" motion (NA: non adaptive, AD: Adaptive)

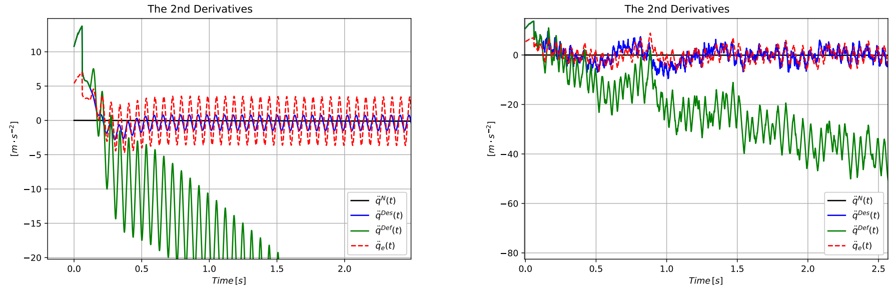

Figure 4.6: Comparison of the development of the adaptive control signal in the noise-free [LHS] and noisy [RHS] cases of slow motion

Finally, Figs. 4.7–4.9 belong to the case of very slow motion, when the adaptive feedback causes greater tracking error than the traditional PID feedback-based Computed Torque Control.

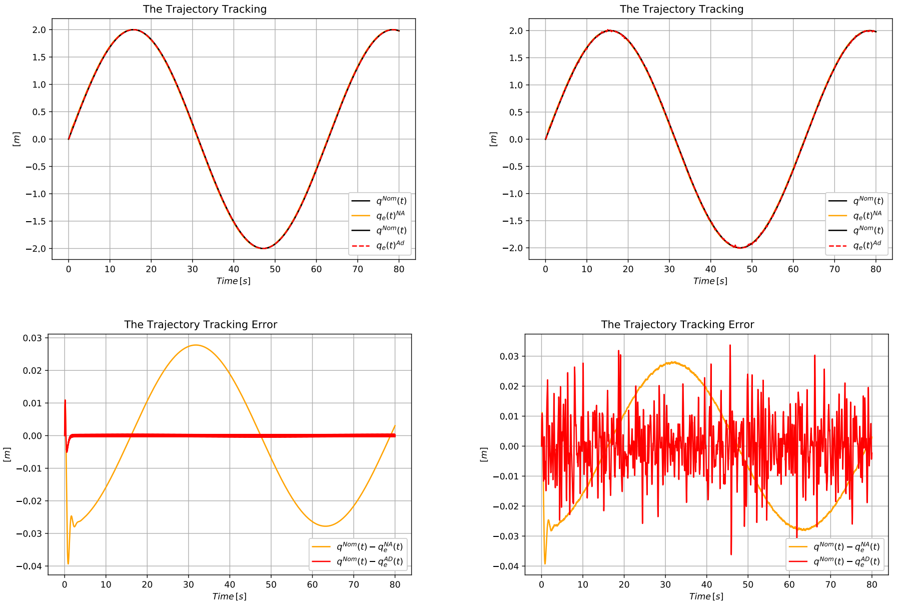

Figure 4.7: Comparison of effect of adaptivity on the: (At top: trajectory tracking) , (At bottom: related trajectory tracking error) in the noise-free [LHS] and noisy [RHS] cases of very slow motion (NA: non adaptive, Ad: Adaptive)

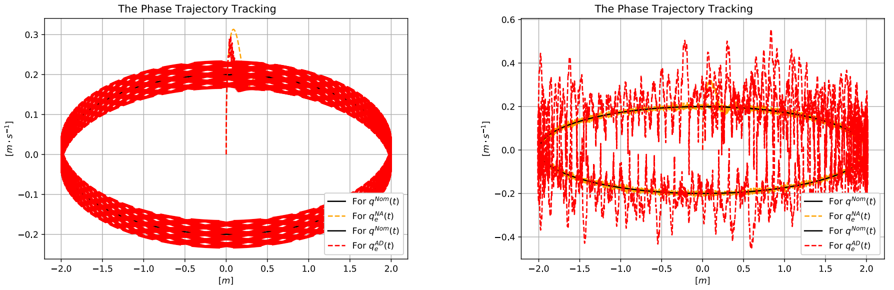

Figure 4.8: Comparison of effect of adaptivity on the phase trajectory tracking in the noise-free [LHS] and noisy [RHS] cases of "very slow" motion (NA: non adaptive, AD: Adaptive)

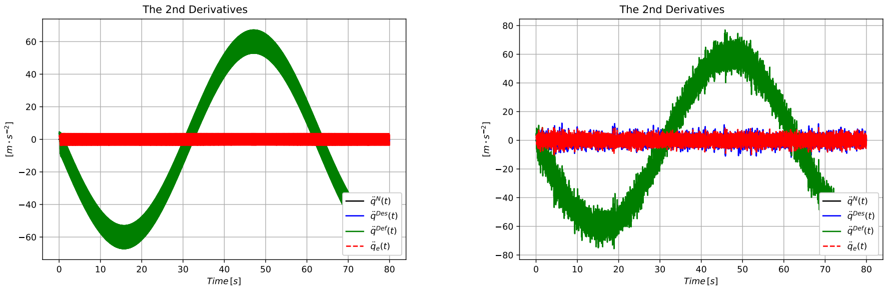

Figure 4.9: Comparison of the development of the adaptive control signal in the noise-free [LHS] and noisy [RHS] cases of very slow motion

To summarize the possible observations, it can be stated as follows:

Typical result is obtained for the trajectory and phase trajectory tracking of fast motion in Figs. 4.1 and 4.2. It is evident that in this case the adaptivity well improved the control quality in the noisy case, too. Figure 4.3 reveals that the noises essentially did not blocked the adaptive algorithm: the desired and the realized values were very close to each other, while the extent of the necessary adaptive deformation was quite considerable.

For the slow motion, the trajectory tracking errors displayed in Fig. 4.4 testify that when the dynamics of the motion to be tracked, due to the extra feedback properties of the FPI-based method, the tracking error of the adaptive motion slightly increased due to the measurement noise. However, it is much better than that of the non-adaptive one. The phase trajectories in Fig. 4.5 also confirm this observation. The zoomed in excerpts of the formation of the control force in Fig. 4.6 testifies that the qualitative trends in the formation of the control signal essentially remained the same in the noise-free and the noisy cases.

Finally, Figs. 4.7 and 4.8 reveal that for very slow motion, i.e., for keeping the system in a static position, application of the adaptive control makes not sense because it feeds back unnecessary noise components. The second time-derivatives of the generalized coordinates described in Fig. 4.9 testify that the adaptive deformation also takes part in the compensation of the consequences of the almost static modeling errors. The nominal $\ddot{q}^N$ values are almost the same as the desired ones ($\ddot{q}^{Des}$), i.e., no significant PID corrections were added to the nominal motion, however, the deformed $\ddot{q}^{Def}$ quite considerably differ from the nominal and the desired values.

The above observations correspond to the expectation that the adaptive approach was invented for the compensation of the effects of the modeling errors in the dynamic model of the system. In the case of slow motion the effects to be compensated by the adaptivity, with the exception of the static terms, simply do not play any role (they provide little contribution in the equation of motion according to the dynamic model), and in this case the usual, classic PID-type feedback can work almost perfectly. The consequences of modeling errors of static nature can be well compensated for by the simple non-adaptive approach, too.

4.5 Comparison of Noise Filtering between Fixed Point Iteration-based Adaptive Control and Unscented Kalman Filter

In control applications the use of observed noisy and sometimes incomplete sets of observations makes a general problem arise that traditionally is tackled by the use of various Kalman filters. The main point behind these filters is to provide some optimized output based on the assumption that the measurement noises are of Gaussian nature and that the subsequent measurements of the same variables are statistically independent. The original concept was developed for linear system model, the later variants were extended to tackle nonlinear models, too, since the nonlinearities make the observation and filtering problems even more significant than in the case of linear systems. The Fixed Point Iteration-based adaptive controllers are especially noise-sensitive since they need the feedback of higher order derivative errors than the usual Resolved Acceleration Rate controllers. For supporting them simpler noise filtering techniques were developed than the Kalman filters since no filter optimization issues were considered by them. In the present investigations the operation of various noise filtering methods are compared with each other in this special control applied for the strongly nonlinear van der Pol oscillator. The simulations confirm that instead of the use of complicated Kalman filter the simpler ones seem to be applicable as well. The scheme of the adaptive controller was outlined in Fig. 3.1.

4.5.1 The Compared Noise Filtering Methods

I. Filtering with Affine Signal Approximation

In the investigations the *observed noisy signal* denoted by $q_o(t)$ and the realized one $q(t)$ was used for the computation of the tracking error components defined in (4.8).

$$e_{int}(t) := \int_{t_0}^{t} (q_o(\xi) - q(\xi))\,\mathrm{d}\xi, \quad \left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t}\right)^3 e_{int}(t) \equiv 0 \tag{4.8}$$

This statement is valid for the *Filtering with Affine Signal Approximation* and the *Application of an Efficient Third Order Low Pass Filter* parts.

This affine function approximation technique was used and described in Section 4.2. During the simulations, the control parameters were set as follows: $\tau = \delta t = 10^{-3}\,[\mathrm{s}]$, $w = 6\,[\mathrm{m}\cdot\mathrm{s}^{-2}]$, $A = -0.22\,[\mathrm{s}^{-2}]$ (in (4.4)), and $\Lambda_{FPI-1} = 8.0\,[\mathrm{s}^{-1}]$ (in (4.8)). The length of the digital filter was $L = 30$.

II. Application of an Efficient Third Order Low Pass Filter

This method was described in section 4.1. In the block *Kinematic Needs* of Fig. 3.1 the appropriate exponent in (4.8) was $\Lambda_{FPI-2} = 12.0[\mathrm{s}^{-1}]$, while the filtering constant λ or $\lambda_{FPI-2} = 250.0[\mathrm{s}^{-1}]$ in (4.1) was in use. The *Fixed Point Transformation Function* applied in the block *Deformation* of Fig. 3.1 in this case was the original *Robust Fixed Point Transformation* published in [84] as in (3.14) in which the same sigmoid function was in use as in (4.4), and in the simulations $K_c = 10^6$ [m $\cdot$ s^{-2}], $B_c = -1$, and $A_c = \frac{5\times10^{-2}}{K_c}$ [m$^{-1}\cdot$ s^2] were applied.

III. Unscented Kalman Filter

Unscented Kalman Filter is considered one of the most widely spread statistical techniques which can be used for estimation e.g., in financial applications, nonlinear control systems, and many more subject areas wherever the full state feedback is required. The filter is based on the nonlinear *Unscented Transformation* that calculates the statistical properties for an arbitrary random variable [64, 65, 110, 111]. In the applications the nonlinear system is described as function of firstly the input state variables q_k which is the dynamic description of the studied system in addition to which the process noise w_{k-1} is considered, and secondly, the observed values or the system measurements z_k to which also is added a so called *sensor noise* v_k in (4.9)

$$q_k = f(q_{k-1}, u_{k-1}) + w_{k-1} \ , \tag{4.9a}$$

$$z_k = h(q_k) + v_k \ . \tag{4.9b}$$

In our calculations a strictly causal process was assumed with $w_k = 0$, furthermore it was assumed that the variables are directly measurable, i.e., $z_k \equiv h(q_k) = q_k + v_k$ was assumed with an appropriate measurement noise. For the discretized dynamic model (i.e., for the first equation of (4.9)) the approximation

$$\begin{aligned} \dot{q}_j &\approx \dot{q}_{j-1} + \frac{\delta t}{m}\left(-kq_j - b(q_j^2 - a^2)\dot{q}_j + u_j\right) \ , \\ q_j &\approx q_{j-1} + \delta t \dot{q}_j \ , \end{aligned} \tag{4.10}$$

that was created on the basis of the dynamic model of the controlled system in (4.21) (a van der Pol oscillator). In the case of the *Extended Kalman Filter (EKF)*, in general, the first order Taylor series approximation of the functions in (4.9) is applied that is rather appropriate to tackle slight nonlinearities. In [63] a new idea was introduced according to which instead using the Jacobians of the functions in (4.9), the state distribution was specified by a minimal set of deterministically chosen sample points of a Gaussian random variable (i.e., the sigma points), and these points were propagated through the nonlinear system. The so obtained points better approximate the posterior mean and covariance than the method based on the Jacobians.

In this study, the state vector physically is $q(k) = [\dot{q}_k, q_k]$ with the initial conditions of the mean is $\hat{q}_0 = E[q_0]$ and the initial covariance $P_0 = E[(q_0 - \hat{q}_0)(q_0 - \hat{q}_0)^T]$. In every cycle the sigma points of the probability density distribution (it is tacitly assumed to be Gaussian or Normal distribution) are calculated for $L = 2$ as

$$\begin{aligned} \chi_{k-1} = \Big[&\hat{q}_{k-1}, \hat{q}_{k-1} + \sqrt{(1+\lambda)P_{k-1}}, \hat{q}_{k-1} + \sqrt{(2+\lambda)P_{k-1}}, \\ &\hat{q}_{k-1} - \sqrt{(1+\lambda)P_{k-1}}, \hat{q}_{k-1} - \sqrt{(2+\lambda)P_{k-1}}\Big] \end{aligned} \tag{4.11}$$

where $\hat{q}_{k-1}$ is a matrix of size 2×1, and the sigma points are obtained by adding and subtracting to it the appropriate columns of the *Lower Triangle Cholesky Factorization Matrix*

[112]. The definition of the parameters can be found in table 4.2. Then, the dynamic model $f(q_k, u_k)$ is evaluated via sigma points as $\chi^*_{k|k-1} = f(\chi_{k-1}, u_{k-1})$. Then the values of estimated prior state and covariance are computed by multiplying their values by the weighted sample means W_i^m and W_i^c in (4.12), respectively (their values are given in table 4.2)

$$\hat{q}_k^- = \sum_{i=0}^{2L} W_i^m \chi^*_{i,k|k-1} \tag{4.12a}$$

$$P^-_{k|k-1} = \sum_{i=0}^{2L} W_i^c [\chi^*_{i,k|k-1} - \hat{q}_k][\chi^*_{i,k|k-1} - \hat{q}_k]^T + Q_{Noise}. \tag{4.12b}$$

The process noise covariance $Q_{Noise} \in \mathbb{R}^{2\times 2}$ where it is set to be $[10^{-1}; 10^{-3}]$.

Table 4.2: The UKF Scaling Parameters

Parameter	***Description***	***Value***
α	Primary Scaling	10^{-2}
β	Secondary Scaling	10^{-3}
κ	Scalar	0.0
L	State vector Dimension	2
λ	Scalar	$\alpha^2(L+\kappa)-L$
W_0^m	Initial State Weight	$\frac{\lambda}{L+\lambda}$
W_i^m	State Weight	$\frac{1}{2(L+\lambda)}$
W_0^c	Initial Covariance Weight	$\frac{\lambda}{L+\lambda}+(1-\alpha+\beta)$
W_i^c	Covariance Weight	$\frac{1}{2(L+\lambda)}$
Λ_{UKF}	Positive constant in(4.8) $[s^{-1}]$	36.0

After that, UKF starts the correction phase for $L = 2$ as

$$\begin{aligned} \chi_{k|k-1} := \Big[& \hat{q}^-_{k-1}, \hat{q}^-_{k-1} + \sqrt{(1+\lambda)P^-_{k-1}}, \hat{q}^-_{k-1} + \sqrt{(2+\lambda)P^-_{k-1}} \\ & \hat{q}^-_{k-1} - \sqrt{(1+\lambda)P^-_{k-1}}, \hat{q}^-_{k-1} - \sqrt{(2+\lambda)P^-_{k-1}} \Big]. \end{aligned} \tag{4.13}$$

Then the unscented transformation is done over the observed values i.e., the sigma points are calculated by the function of $\gamma_{k|k-1} = h(\chi_{k|k-1})$ and then recombined to produce the predicted measurement values $\hat{z}_k^- = \sum_{i=0}^{2L} = W_i^m \gamma_{i,k|k-1}$ and the predicted measurement covariance

$$P_{z_k^- z_k^-} = \sum_{i=0}^{2L} W_i^c [\gamma_{i,k|k-1} - \hat{z}_k^-][\gamma_{i,k|k-1} - \hat{z}_k^-]^T + R_{Noise} \tag{4.14}$$

where the measurement noise covariance $R_{Noise} \in \mathbb{R}^{2\times 2}$ were set to $[10^{-2}; 10^{-3}]$. The trading values between the state and measurement are obtained by calculating the cross covariance in (4.15)

$$P_{q_k z_k} = \sum_{i=0}^{2L} W_i^c [\chi_{i,k|k-1} - \hat{q}_k^-][\gamma_{i,k|k-1} - \hat{z}_k^-]^T \tag{4.15}$$

that allows to compute the Kalman gain $K_{gain} = P_{q_k z_k} P_{q_k z_k}^{-1}$, and thus to obtain the updated values of the state variables and covariance by:

$$\hat{q}_k = \hat{q}_k^- + K_{gain} \left(\hat{z} - \hat{z}^- \right) \tag{4.16a}$$

$$P_k = P_k^- - K_{gain} P_{z_k^- z_k^-} K_{gain}^T \quad . \tag{4.16b}$$

In general it is not easy to find appropriate values for Q_{Noise}, and R_{Noise}. In the literature generally small (not zero) values are recommended the effects of which spread toward (4.12) and (4.14). For making the most possible correct comparison their values were experimentally set so that the best behavior of the Kalman filter was achieved.

4.5.1.1 General Assumptions Regarding the Inherited Noise Distribution in UKF

For the first derivative of the coordinate value the following assumptions were considered and utilized:

- Let x_i denote the actual coordinate value at time instant t_i and $\hat{x}_i := x_i + \mu_i$ its noisy measured value later used for numerical differentiation. It is assumed that $\forall t_i$ the probability density distribution of the additive noise component is $\varphi(\mu)$.

- Then by definition

$$E(\mu) := \int \mu \varphi(\mu) \mathrm{d}\mu \quad , \tag{4.17a}$$

$$\sigma^2(\mu) := \int \varphi(\mu)(\mu - E(\mu))^2 \mathrm{d}\mu \quad , \tag{4.17b}$$

$$\int \varphi(\mu) \mathrm{d}\mu = 1 \quad . \tag{4.17c}$$

  where the assumption $E(\mu) = 0$ is reasonable, and it leads to $\sigma^2(\mu) = \int \varphi(\mu)\mu^2 \mathrm{d}\mu$.

- Let $\psi(\mu_i, \mu_{i-1})$ the probability density distribution of the measurements made in time instants t_i and t_{i-1}. If δt is the time-resolution of the discrete differentiation then the mean of the velocity computed from the measured values will be

$$\begin{aligned} & E\left(\frac{q_i - q_{i-1}}{\delta t}\right) = \\ & = \int\int \psi(\mu_i, \mu_{i-1}) \frac{x_i + \mu_i - x_{i-1} - \mu_{i-1}}{\delta t} \mathrm{d}\mu_i \mathrm{d}\mu_{i-1} \quad , \\ & \int\int \psi(\mu_i, \mu_{i-1}) \mathrm{d}\mu_i \mathrm{d}\mu_{i-1} = 1 \quad . \end{aligned} \tag{4.18}$$

- For independent measurements $\psi(\mu_i, \mu_{i-1}) = \varphi(\mu_i)\varphi(\mu_{i-1})$ it yields

$$\begin{aligned} & E\left(\frac{x_i - x_{i-1}}{\delta t}\right) = 1\frac{x_i - x_{i-1}}{\delta t} + 1E\left(\frac{\mu_i}{\delta t}\right) \\ & - 1E\left(\frac{\mu_{i-1}}{\delta t}\right) = \frac{x_i - x_{i-1}}{\delta t} \quad . \end{aligned} \tag{4.19}$$

- For the calculation of the standard deviation it can be stated that

$$\begin{aligned}&\int\int \psi(\mu_i,\mu_{i-1})\left(\frac{x_i+\mu_i-x_{i-1}-\mu_{i-1}}{\delta t}\right.\\&\left.-\frac{(x_i-x_{i-1})}{\delta t}\right)^2 \mathrm{d}\mu_i \mathrm{d}\mu_{i-1}=\\&=\int\int \psi(\mu_i,\mu_{i-1})\left(\frac{\mu_i-\mu_{i-1}}{\delta t}\right)^2 \mathrm{d}\mu_i \mathrm{d}\mu_{i-1}\\&=\int\int \varphi(\mu_i)\varphi(\mu_{i-1})\frac{\mu_i^2+\mu_{i-1}^2-2\mu_i\mu_{i-1}}{\delta t^2}\mathrm{d}\mu_i \mathrm{d}\mu_{i-1}\\&=2\frac{\sigma^2(\mu)}{\delta t^2}\end{aligned} \tag{4.20}$$

It must be noted that in the case of the first two methods the process noise was not interpreted, only the observation noise had physical meaning, i.e., the process was considered strictly causal.

4.5.2 Simulation Results

The simulations were made in *Julia language* by discrete time resolution $\delta t = 10^{-3}$ [s] and a sinusoidal nominal trajectory has the function $q^N(t) = Ampl\sin(\omega t) + Shift$ in which the values of the noise amplitude were 10^{-5}[m] and $10^{-4}[m]$, and the signal amplitude and the shift were $Ampl = 0.5$ [m], $Shift = 0.08$ [m], respectively. For illustrating the compared results, the van der Pol oscillator was chosen with its dynamic equation of motion in (4.21)

$$m\ddot{q} + b(q^2 - a^2)\dot{q} + kq = u \tag{4.21}$$

and the used parameters are given in table 4.3.

Table 4.3: The System Parameters

Parameter	*Exact Value*	*Approx Value*
Mass m [kg]	1.0	1.3
Spring stiffness k [N $\cdot$ m^{-1}]	100.0	120.0
Viscous damping b [N $\cdot$ m^{-3} $\cdot$ s]	0.6	0.8
Separator a $[m]$	1.5	1.7

4.5.2.1 Comparisons Without Noise

As it was expected, the results that are illustrated in Figs. 4.10,4.11,4.12 show that the UKF was powerful in the ideal situation i.e., without any noisy addition, however, the responses of the first FPI method had an acceptable trajectory tracking error in a range of ±0.6% while the value of its first derivative tracking error had a tremendous increase in a range of ±20%. The second FPI method produced better results in both the trajectory and first derivative tracking, where the error was in a range of ±0.1% and ±0.5% respectively.

The best results belong to UKF with trajectory tracking error of 10^{-4} order of magnitude with $\pm 0.6\%$ for first derivative tracking error.

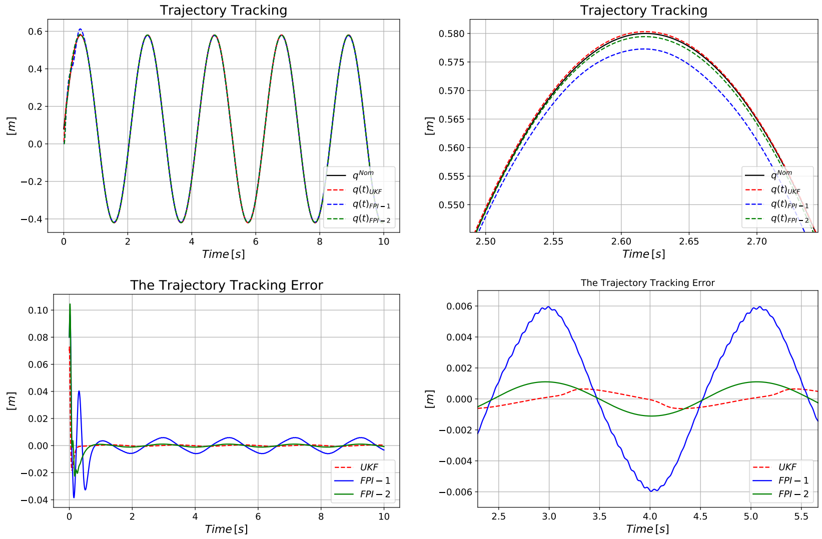

Figure 4.10: Comparison of the three methods without noise: at the top: trajectory tracking, at the bottom: trajectory tracking error.

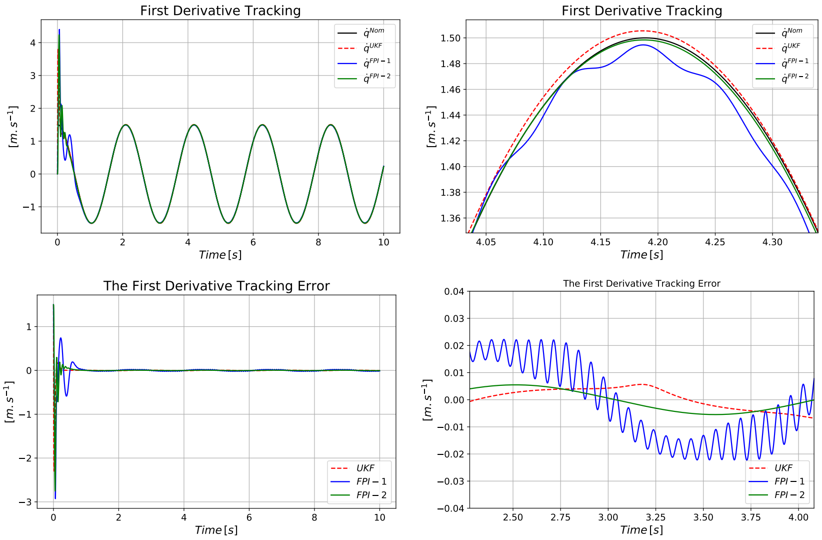

Figure 4.11: Comparison of the three methods without noise: at the top: first derivative tracking and its zoomed response, at the bottom: first derivative tracking error and its zoomed response.

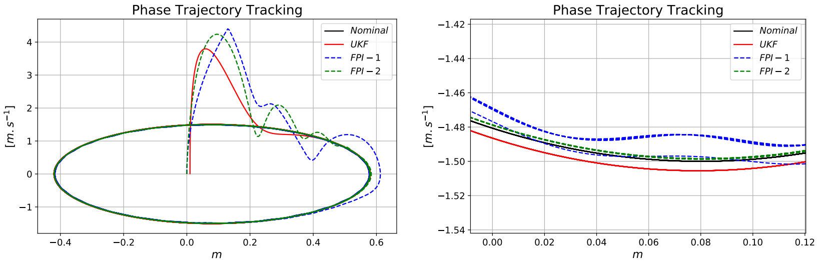

Figure 4.12: Comparison of the three methods without noise: phase trajectory tracking and its zoomed response.

For the noise-free case for the results of the three methods the floowing observations can be done:

- The trajectory tracking errors displayed in te bottom of Fig. 4.10 testify that the best result was produced by the UKF, the FPI-2 method produced comparable precision, while the FPI-1 approach produced the highest tracking error.
- Figure 4.11 reveals that the same can be said about the first time-derivative of the tracking errors.
- Figure 4.12 describing the phase trajectories confirms the above observations.

4.5.2.2 Comparison With Gaussian Distribution Noise

The noise amplitude firstly was examined with $10^{-5}[m]$ as in Figs. 4.13,4.14,4.15. It can be noticed that due to the small amplitude, the two FPI-based methods effectively filtered the noise. The UKF produced small spikes in its response that have very small amplitude and correspond to stable conditions. After that, the noise amplitude was increased to $10^{-4}[m]$ where growing differences were observed. It became obvious that the FPI-2 showed the best stability with minimum trajectory tracking error and first derivative tracking error (Figs. 4.16,4.17,4.18).

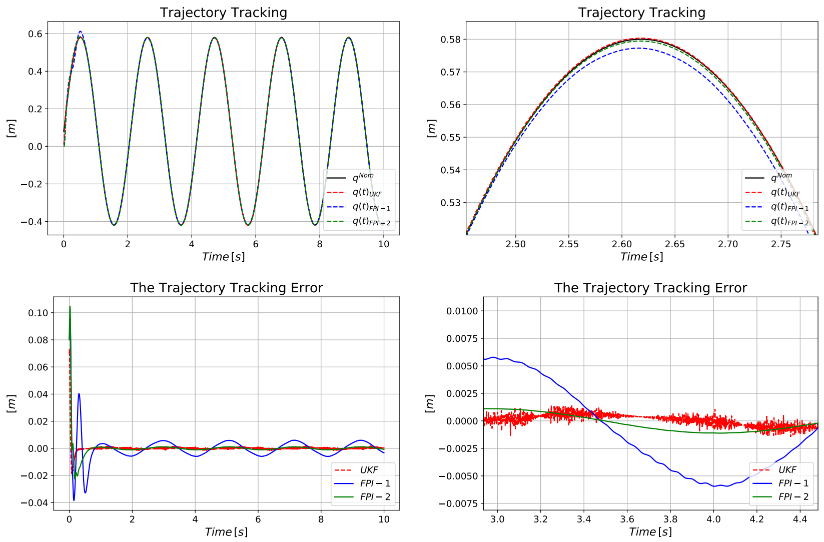

Figure 4.13: Comparison of the three methods in the case of noise amplitude = 10^{-5} $[m]$: at the top: trajectory Tracking, at the bottom: trajectory tracking error.

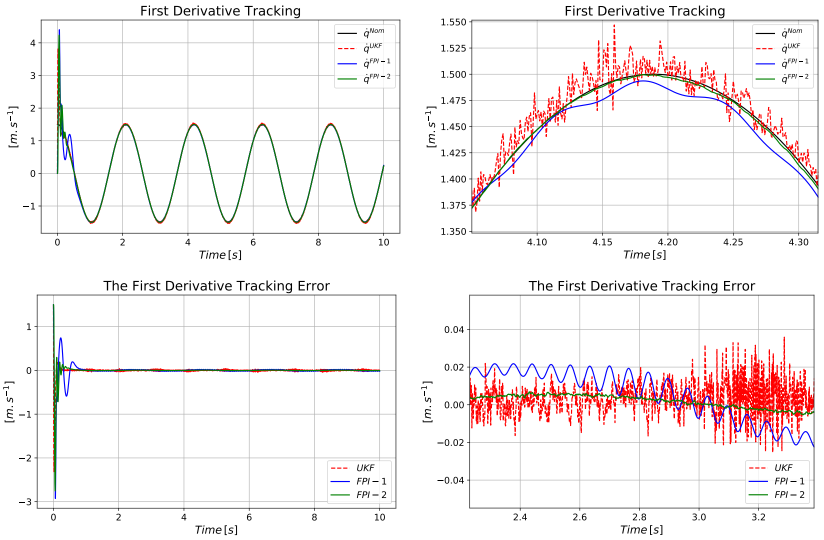

Figure 4.14: Comparison of the three methods in the case of noise amplitude = 10^{-5} $[m]$:at the top: first derivative tracking and its zoomed response, at the bottom: first derivative tracking error and its zoomed response.

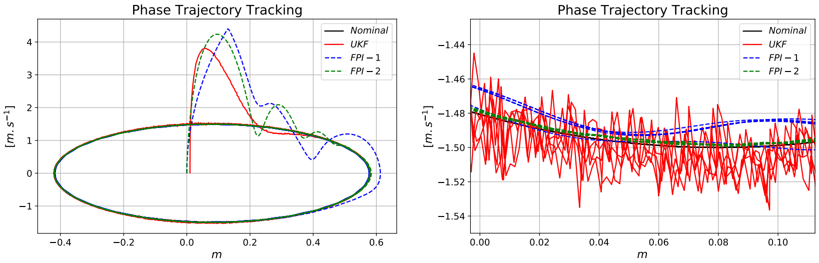

Figure 4.15: Comparison of the three methods in the case of noise amplitude = $10^{-5}\,[m]$: phase trajectory tracking and its zoomed response.

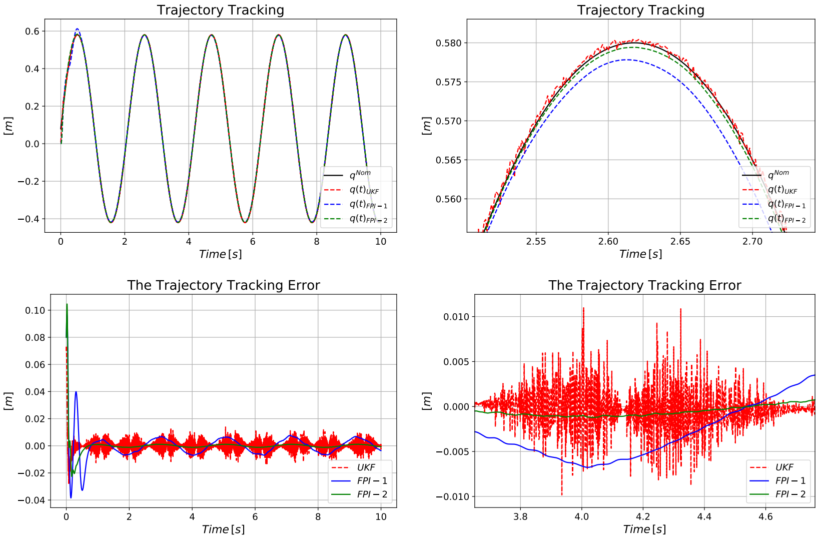

Figure 4.16: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: at the top: trajectory tracking, at the bottom: trajectory tracking error.

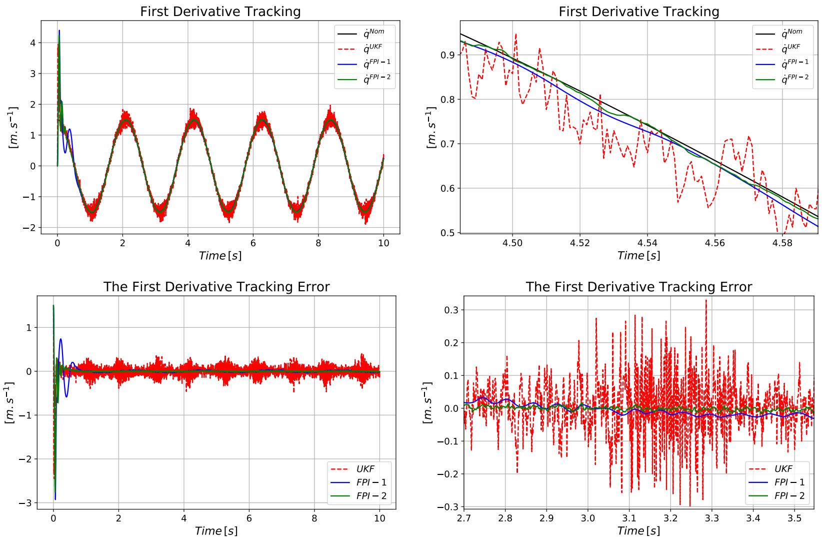

Figure 4.17: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: at the top: first derivative tracking and its zoomed response, at the bottom: first derivative tracking error and its zoomed response.

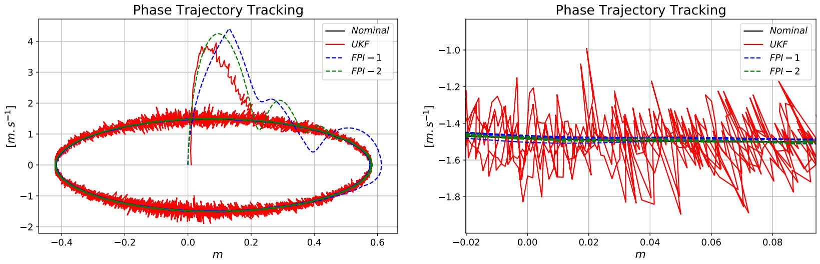

Figure 4.18: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: phase trajectory tracking and its zoomed response.

When the noise amplitude was increased to $10^{-4}[m]$ the UKF produced so great noise that the other two methods produced considerably better solutions because their smoothness (Fig. 4.16).

For the comparison the effects of measurement noises the following statements can be summarized:

- The small noise amplitude ($10^{-5}[m]$) resulted in the same order of precision of trajectory tracking as in the noise-free case (Fig. 4.13): the best results were provided by the UKF, the second best was the FPI-2 solution, and FPI-1 produced geater tracking error. However, the signal of the UKF was more noisy than the signal of the other two methods. When the noise amplitude was increased to $10^{-4}[m]$ the UKF produced so great noise that the other two methods produced considerably better solutions because their smoothness (Fig. 4.16).

- In the first coordinate derivatives the UKF produced the most noisy signal. The other two methods yielded smooth signals (Fig. 4.14). The increased noise amplitude $(10^{-4}[m])$ made the UKF provide very large noise (Fig. 4.17).
- Again, the phase trajectories in Figs. 4.15 and 4.18 confirm the above statements.

4.5.2.3 Comparison With Logistic Distribution Noise

Figures. 4.19,4.20,4.21 answer the interesting question of the effect of changing the Gaussian noise distribution into Logistic. Unlike the UKF, the two FPI methods maintained their ability to reduce the noisy effects. The error of trajectory tacking was in a range of $\pm 0.75\%$ for the first FPI method against $\pm 0.12\%$ for the second one while the first derivative tracking error was in a range of $\pm 4\%$ for the $FPI-1$ and $\pm 2.5\%$ for the $FPI-2$.

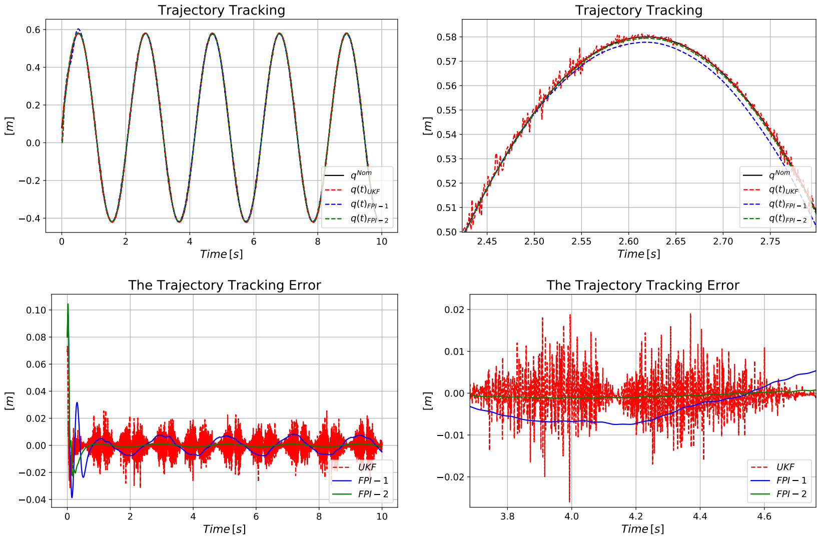

Figure 4.19: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: at the top: trajectory tracking, at the bottom: trajectory tracking error.

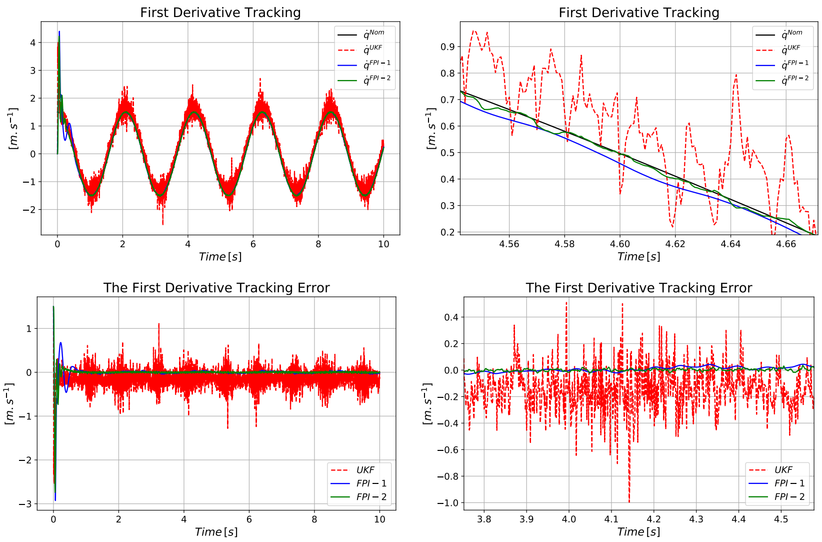

Figure 4.20: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: at the top: first derivative tracking and its zoomed response, at the bottom: first derivative tracking error and its zoomed response.

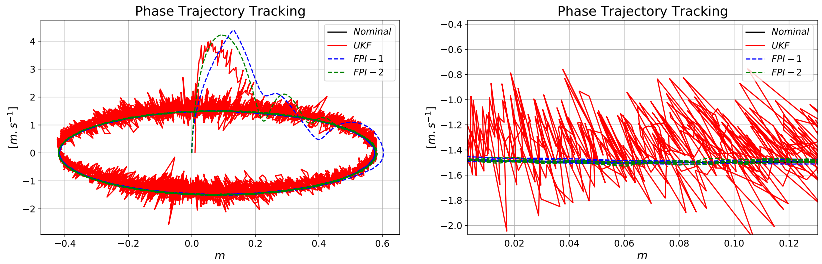

Figure 4.21: Comparison of the three methods in the case of noise amplitude = $10^{-4}\,[m]$: phase trajectory tracking and its zoomed response.

Figures. 4.19, 4.20, and 4.21 made for the noise amplitude $10^{-4}\,[m]$ for the logistic noise distribution prove that the UKF achieved the worst result. This fact is not surprising because the UKF is optimized for Gaussian noise distribution.

For showing the differences of responses for the three methods, the Mean Absolute Error (MAE) and Mean Squared Error (MSE) errors were used. The initial transient parts, in which the errors characterize the differences in the initial conditions of the nominal trajectories and the realized ones, were omitted in the calculations. The results are described in Table. 4.4 for trajectory tracking error and Table. 4.5 for trajectory tracking derivative error. The tables reveal that for small noise amplitudes the UKF yielded the best solution for trajectory tracking. When a Gaussian noise of higher amplitude appeared, the method FPI-2 made better result than the UKF. Furthermore, when the noise distribution was changed form Gaussian (to which the UKF was optimized), the FPI-2 method gave much better solution than the UKF. With regard to the velocity tracking error, the UKF was yielded comparable result with the FPI-2 method only in the lack of noises. For higher noise amplitude the UKF became more sensitive, and for the Logistic distribution it yielded considerably worst result than the FPI-2 (the best) and the FPI-1 method. I note that these features can be seen in the figures qualitatively.

Table 4.4: For trajectory tracking error

Noise Type	**MAE**			**MSE**		
	$FPI1$	$FPI2$	UKF	$FPI1$	$FPI2$	UKF
No noise	0.003477	0.00069472	0.00036234	1.538e-05	6e-07	1.7e-07
Gaussian 10^{-5} $[m]$	0.0034724	0.00069474	0.00041416	1.535e-05	6e-07	2.4e-07
Gaussian 10^{-4} $[m]$	0.00388562	0.0006954	0.0018963	1.908e-05	6e-07	7.24e-06
Logistic 10^{-4} $[m]$	0.00443953	0.00069657	0.00326782	2.463e-05	6.1e-07	2.302e-05

Table 4.5: For trajectory tracking derivative error

Noise Type	**MAE**			**MSE**		
	$FPI1_p$	$FPI2_p$	UKF_p	$FPI1_p$	$FPI2_p$	UKF_p
No noise	0.01281402	0.00347075	0.00355144	0.00020169	1.491e-05	1.554e-05
Gaussian 10^{-5} $[m]$	0.01282307	0.00348547	0.00729481	0.00020182	1.516e-05	9.055e-05
Gaussian 10^{-4} $[m]$	0.01401114	0.00511788	0.06556157	0.00026439	4.066e-05	0.0074151
Logistic 10^{-4} $[m]$	0.0161083	0.00839336	0.17379961	0.00036574	0.00010992	0.04516468

4.6 Thesis Statement III

By the use of simulation results I have shown that the FPI-based adaptive technique can well cooperate with the simple noise filtering techniques as the cascade of moving windows with affine signal approximation and the simple low pass filter, that do not apply any assumption on the statistical nature on the measurement noises. I have also shown that it is difficult to combine this method with the more traditional UKF that was found less effective. As possible reason I identified the fact that the mathematical structure of the UKF cannot be so well fitted to that of the FPI-based adaptive approach.

Related own publications: **[A. 7], [A. 8] and [A. 9].**

Chapter 5

Implementing Fixed Point Iteration-based Adaptive Control and Particle Swarm Optimization

In comparison with the classical Lyapunov function-based parameter identification-based techniques, the advantage of the fixed point iteration-based techniques is that they do not require the identification of the model parameters. However, for other purposes, especially in life sciences, certain parameter values have definite significance and their identification is required independently of solving the control task. This naturally make the question arise if is it possible to identify the system's parameters (or at least certain parameters of the system) while the FPI-based adaptive controller is in operation. It is well known that the early adaptive controllers in robotics [113] the dynamic model of the controlled system was so formulated that an array of the identifiable dynamical system parameters were separated from an array of kinematically precisely known components, and the approximate system parameters were slowly tuned by Lyapunov functions-based constructions. At beginning of the tuning process this method worked with large trajectory tracking error since it always used an only slowly improved parameter set for the calculation of the necessary control forces. Later Dineva combined this learning method with the FPI-based adaptive controller in [114–116]. The main idea was that the FPI-based control signal was used for realizing the adaptive control, but instead of the original PID-type feedback terms used by Slotine and Li, she utilized the FPI-based terms for parameter tuning. This method resulted precise tracking from the beginning, and the use of the improved parameters in the approximate model in the adaptive control was only optional. In this research I investigated a similar problem: while the adaptive controller worked on an FPI basis, for the identification of the system parameters more general, evolutionary computation-based methods could be used. Utilization of the improved parameters in the adaptive control remained rather a free option than a must, as well as in the case of Dineva's investigations. For the simulations I have chosen a strongly nonlinear model with limited number of parameters, and a particular evolutionary method, the simple and elegant Particle Swarm Optimization method [26].

The first part investigates the limitations of the applicability of Particle Swarm Optimization (PSO) with Fixed Point Operation-based adaptive controllers in case of on-line mode. For this purpose a simple but strongly nonlinear dynamic model was chosen by the use of which the problems of the parameter identification tasks can be well illustrated, expounded, and understood. Especially the limitation of this method was pointed out when the identified model in certain steps replaced the originally used initial one (this is the meaning of the term on-line mode).

In the second part I made studies on the off-line combination of Fixed Point Iteration-based adaptive controllers with (PSO) using a more complicated, realistic robot model. In this approach the FPI-based adaptive controller is used for tracking a nominal trajectory while the (PSO) used to refine the model, without using the improved model in the control.

5.1 Limitations in the Usage of Fixed Point Iteration-based Adaptive Control and Particle Swarm Optimization Illustrated by a Simple Paradigm

The goal is the identification of the *model parameters* of the controlled system during the operation of and FPI-based controller. In this case the tuning strategy can be based on more general learning methods. Since setting the exact model parameters mathematically can be expressed as finding the zero value as absolute minimum of a cost function, various possibilities are available. In [117] the efficiency of finite element methods and derivative-free approaches are compared in the case of singular systems. If it is known *a priori* that the optimization task is *convex*, very efficient special solution tools can be applied. If not, quite general approach is needed. Due to its simplicity the (PSO) was chosen.

For the purpose of the investigations the popular nonlinear benchmark system, the van der Pol oscillator [104], that was a triode that made forced nonlinear oscillations and later its various modifications played important role in the research of chaotic phenomena (e.g., [118]). It was found that our certain expectations were too optimistic, though the really important parameters that played role in bringing about the observed motion were successfully identified.

5.1.1 The Dynamic Model of the van der Pol Oscillator

Though in the practice *the design of an engine*, e.g., that of the water injection system in the compressors of turbo jet engines [119] is based on extremely complex numerical computations, for *the description of the operation of such engine for control technological purposes* a relatively simple model may work well. The van der Pol oscillator –considered as a symbolically mechanical system has the equations of motion as in (5.1) :

$$\ddot{q} = -\frac{k}{m}q - \frac{b}{m}\left(q^2 - a^2\right)\dot{q} + \frac{1}{m}u \tag{5.1a}$$

$$u = m\ddot{q} + kq + b\left(q^2 - a^2\right)\dot{q} \tag{5.1b}$$

in which $k = 250\,[\mathrm{N}\cdot\mathrm{m}^{-1}]$ is a *spring constant*, $m = 10\,[\mathrm{kg}]$ corresponds to the inertia of a mass point, $b = 4.0\,[\mathrm{N}\cdot\mathrm{s}\cdot\mathrm{m}^{-3}]$ excitation/damping parameter, and $a = 2.5\,[\mathrm{m}]$ is a parameter the separates the excitation/damping region from each other. The parameters of the *initial approximate model* used by the controller were $\widehat{k} = 120\,[\mathrm{N}\cdot\mathrm{m}^{-1}]$, $\widehat{m} = 20\,[\mathrm{kg}]$, and $\widehat{b} =$

$2.0\,[\mathrm{N}\cdot\mathrm{s}\cdot\mathrm{m}^{-3}]$, and $\hat{a}=1.5\,[\mathrm{m}]$. Equation (5.1a) reveals that the output of the system model $\ddot{q}$ *depends on some ratios of certain model parameters* as $\frac{k}{m}$, $\frac{b}{m}$, and $\frac{a^2b}{m}$, and the only parameter is $\frac{1}{m}$ the dependence on which has a clear coefficient in the output is the control force u. These relationships make the parameter estimation problem more coupled than estimation of linear regression parameters. If the estimated value of say m is modified, it immediately concerns the effects of the other parameters in the terms $\frac{k}{m}$, $\frac{b}{m}$, and $\frac{a^2b}{m}$. The effects of such inter-dependencies can be identified in the simulation results, too.

It also is trivial that the features of the trajectory to be tracked cannot be separated from the problem of parameter estimation. In different trajectories the variables q, $\dot{q}$, and u have different *significance* in bringing about $\ddot{q}$, therefore it can be expected that different trajectories allow the identification of different model parameters in different manner. The simulation results also underpin this expectation.

5.1.2 Computed Simulations

In the simulations the very simple PSO solution was chosen. The four parameters to be tuned were placed in the first four positions of a row, while in position five the appropriate cost function (the absolute value of the difference between the observed and the estimated $\ddot{q}$ values) were put. Therefore the population of the particles was represented by a matrix the row number of which corresponded to the number of the individual particle, that was described by the values $[k,m,b,a,cost]$. For storing the *best local finding* for a particle a matrix of exactly the same size was used with the rows $[k_{bl},m_{bl},b_{bl},a_{bl},cost_{bl}]$, while the *best global estimation* was stored in a single row as $[k_{bg},m_{bg},b_{bg},a_{bg},cost_{bg}]$. According to the simple and practical principles of PSO, the actual position of the actual particle was *pulled towards* the direction of the *best global particle* by a velocity vector contribution $c_3\mathrm{rand}()[k_{bg}-k,m_{bg}-m,b_{bg}-b,a_{bg}-a]$, and toward the *best local particle* by the vector contribution $c_2\mathrm{rand}()[k_{bl}-k,m_{bl}-m,b_{bl}-b,a_{bl}-a]$ in which the function rand() means even distribution in the interval $[0,1]$. In analogy with the method of *Simulated Annealing* [30], without a sophisticated cooling strategy these contributions were completed with a random term $c_4(rand()-0.5)v$ with $c_4>1$ that was able even to revert the motion of the particle in (5.2).

$$\begin{aligned} v_i(t+1) &= c_1v_i(t)+c_2rand()(BL_i-P_i(t)) \\ &+c_3rand()(BG-P_i(t))+c_4(rand()-0.5)v_i(t)\ . \end{aligned} \tag{5.2}$$

In the traditional terms the particle P_i is pulled toward the local $(\{BL_i\})$ or global $(\{BG\})$ optima, the last term can have an arbitrary direction weighted by the parameter c_4.

The parameters were experimentally set as $c_1=1.0$ (no velocity damping was applied), $c_2=0.2$, $c_3=2.1$, and $c_4=2$ if this term was applied, and $c_4=0$ for the computations without this extra term.

Taking into account that a mathematically correct use of the PSO should evaluate the effect of a given parameter setting $[k,m,b,a]$ for a wide set of various **observed** $\{q,\,\dot{q},\,\ddot{q},\,u\}$ values in which the significance of each parameter is taken from granted, in a real-time control correct estimation cannot be expected. Instead of a wide set of examples of the above space only elements already visited by the system during the control session can be taken into account. Though in the case of a jerk-free continuous motion the subsequent examples are close to each other, the calculation of the cost or the last 12 visited points was a reasonable compromise for finding a sub-optimal estimation of the model parameters. Following 30 digital steps the initial approximate model was replaced with the PSO-based best estimation.

In these simulations the *Robust Fixed Point Transformation*-based method was applied where the values of the control parameters were as follows $\Lambda = 6\,[\mathrm{s}^{-1}]$ from (3.3), and $K_c = 10^6$, $B_c = -1$, and $A_c = \frac{0.1}{K_c}$ from (3.14) were chosen. The time-resolution was $\delta t = 10^{-3}\,[\mathrm{s}]$. Simulations were made for sinusoidal motion with amplitude and frequency similar to the free oscillation of the van der Pol oscillator were considered.

5.1.2.1 Simulations for Sinusoidal Motion

Trajectories for sinusoidal motion around the center points $q_{1_0} = -0.5\,[\mathrm{m}]$, $q_{2_0} = 0.0\,[\mathrm{m}]$, and $q_{3_0} = +0.5\,[\mathrm{m}]$ were investigated. In these computations $c_4 = 0$ was chosen. Figure. 5.1 reveals precise trajectory tracking. According to Fig. 5.2 the phase trajectory is precisely tracked. The jumps in the second time-derivatives reveal that at these points the better model replaced older one, consequently the necessary adaptive deformations abruptly decreased (i.e., the cyan (deformed) signal went back to the vicinity of the desired one), but later increasing adaptive deformation became necessary indicating that certain parameters were not well estimated by the PSO algorithm.

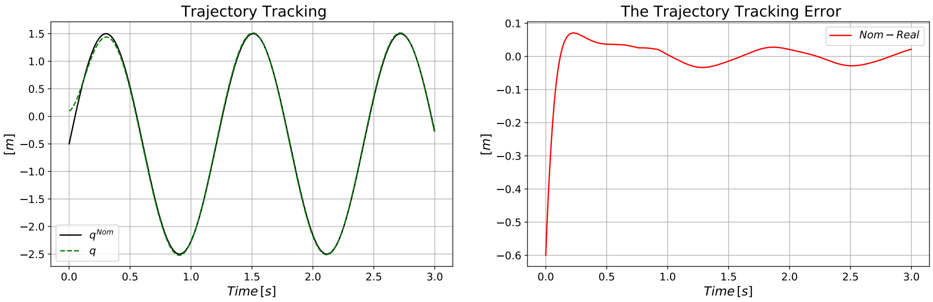

Figure 5.1: The trajectory tracking and its error for sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$

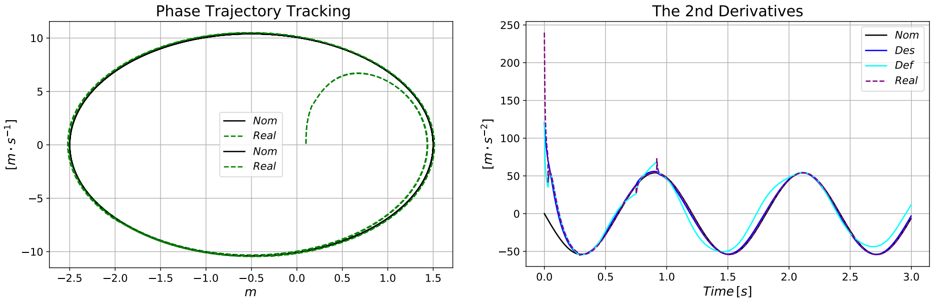

Figure 5.2: The phase trajectory tracking and the 2nd time-derivatives for sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$

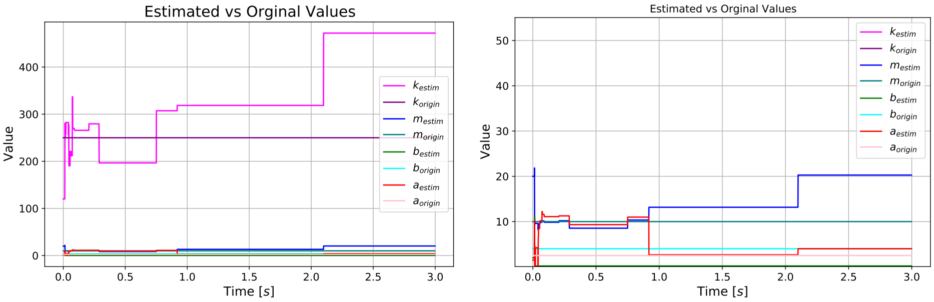

Figure 5.3: The estimated parameters for sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$ (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

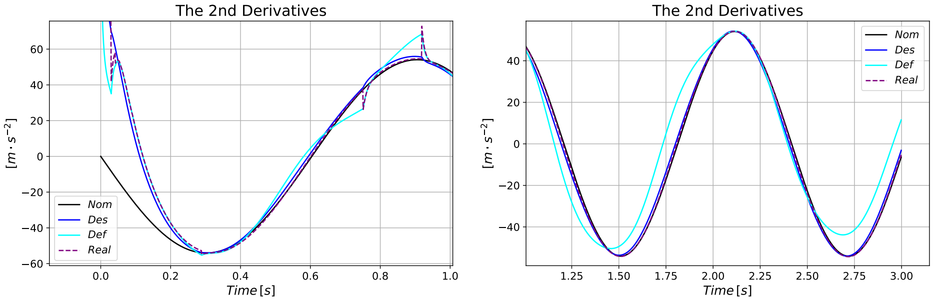

Figure 5.4: The 2nd derivatives for sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$

In the next step the $q_{2_0} = 0.0\,[\mathrm{m}]$ shift values were considered. The precise trajectory tracking witnessed by Fig. 5.5 is accompanied by little adaptive deformations in Fig. 5.6 that indicates that the model was acceptable because its use required only small extent of adaptive deformation. However, this means that only the parameters that are important in the given session of the motion were well identified because Fig. 5.7 indicates high errors in the estimated parameters. The fact that the ratio $\frac{k}{m}$ influences $\ddot{q}$ can be explained if it is taken into account that the parameters k and m were simultaneously overestimated.

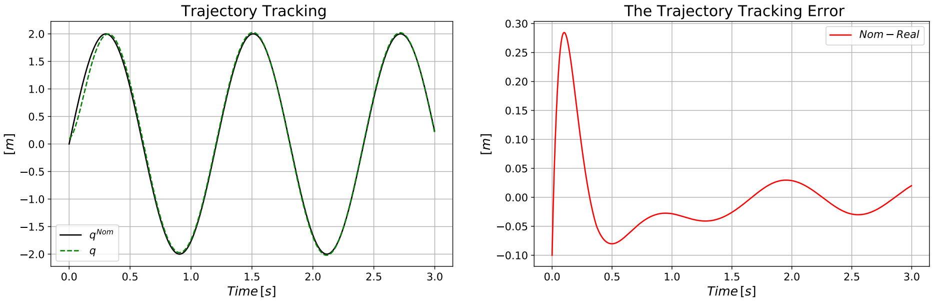

Figure 5.5: The trajectory tracking and its error for sinusoidal motion for trajectory shift of $0.0\,[\mathrm{m}]$

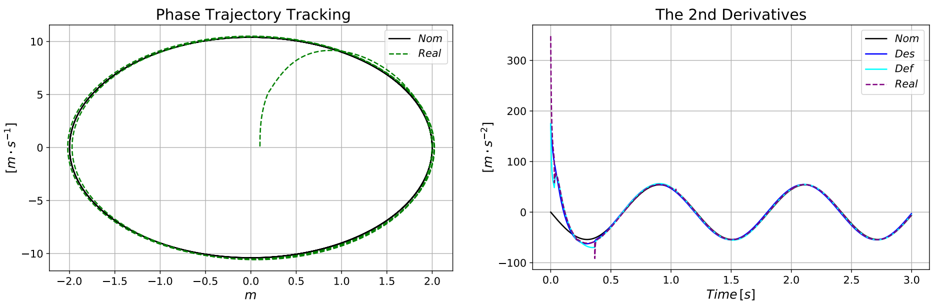

Figure 5.6: The phase trajectory tracking and the 2nd time-derivatives for sinusoidal motion for trajectory shift of 0.0 [m]

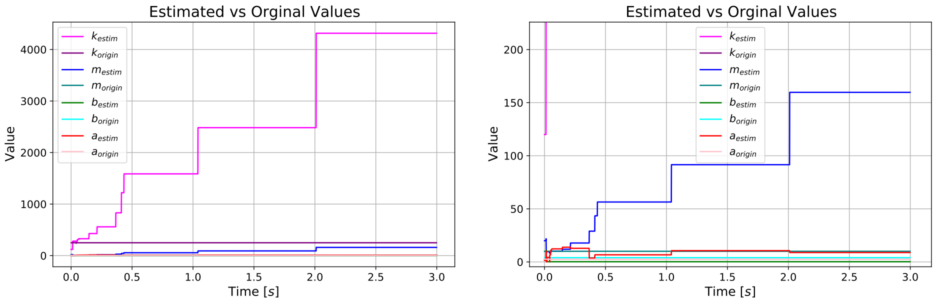

Figure 5.7: The estimated parameters for sinusoidal motion for trajectory shift of 0.0 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

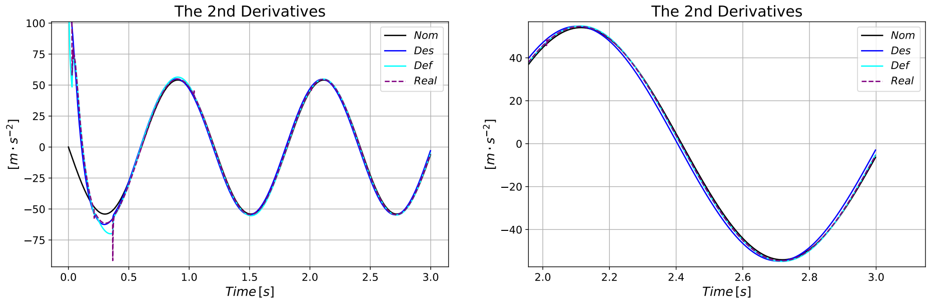

Figure 5.8: The 2nd derivatives for sinusoidal motion for trajectory shift of 0.0 [m]

In the set made for the trajectory shift $q_{3_0} = +0.5$ [m] can be well observed that the precise trajectory tracking (Fig. 5.9) happened with relatively considerable adaptive deformation in Fig. 5.10, though the individual parameters in Fig. 5.11 were better identified.

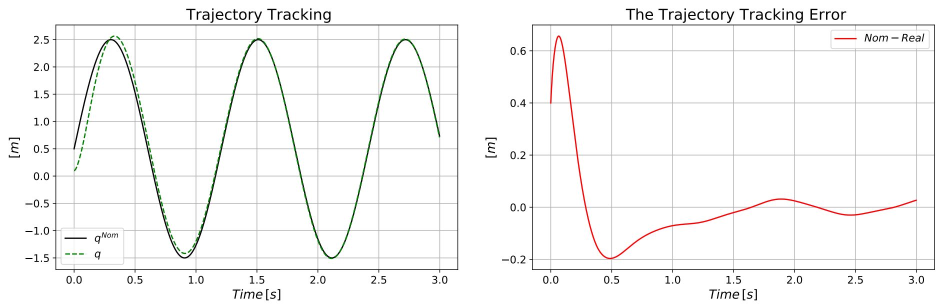

Figure 5.9: The trajectory tracking and its error for sinusoidal motion for trajectory shift of 0.5 [m]

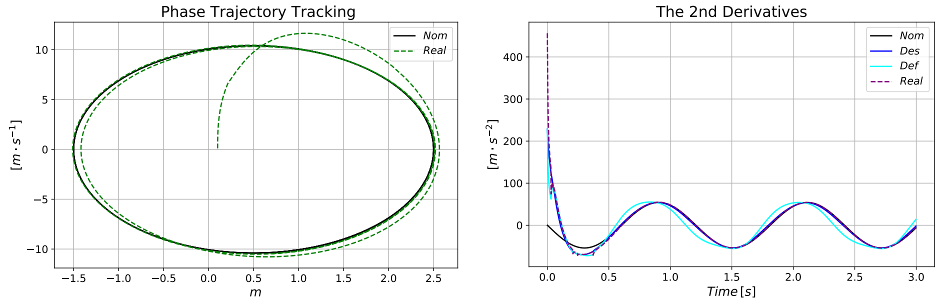

Figure 5.10: The phase trajectory tracking and the 2nd time-derivatives for sinusoidal motion for trajectory shift of 0.5 [m]

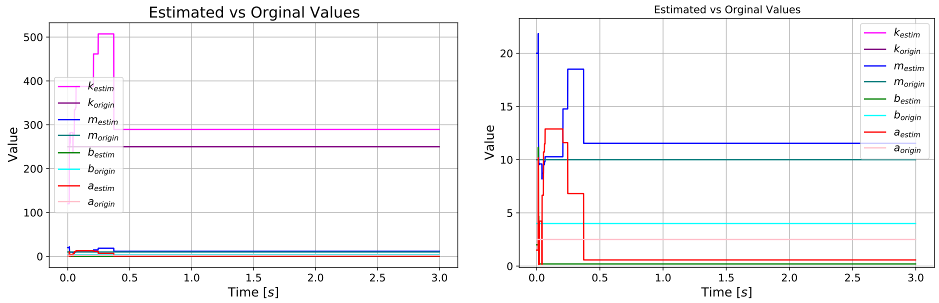

Figure 5.11: The estimated parameters for sinusoidal motion for trajectory shift of 0.5 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

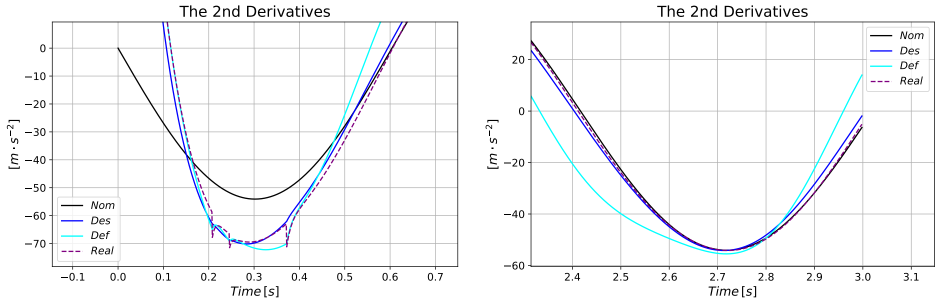

Figure 5.12: The 2nd derivatives for sinusoidal motion for trajectory shift of 0.5 [m]

The observations can be summarized as follows. Due to the special structure of the equation of motion of the van der Pol oscillator (5.1), due to the separating parameter denoted by a, the system's behavior considerably depends on the relationship between $|q|$ and a. For a $q > a$ and $q < -a$ the system is in a damped regime, but is $|q| < a$ it is excited. In the first motion belonging to the shit value $-0.5\,[m]$ described by Fig. 5.1 the coordinate q was placed in the lower region of the excitation ($q \in [-2.5, 1.5]$) with a strong asymmetry in the exciting effect. The zeros shift value placed the nominal trajectory into the center of excitation in a symmetric manner $q \in [-2, 2]\,[m]$. Finally the shift value $0.5\,[m]$ place the motion into the upper part of the excitation in an asymmetric manner $q \in [-1.5, 2.5]\,[m]$. The appropriate Figs. 5.1, 5.5, and 5.9 testify the following the greater transient error, the tracking error was stabilized approximately in the interval $[-0.05, 0.05]$ in each case. Since the method worked online, and utilized the parameter values of the best estimation, in Figs. 5.2, 5.6, and 5.10 certain jumps can be observed in the graphs of the second time-derivatives that quickly relaxed and did not have observable effects in the phase trajectories. The jumps in the identified parameters are described in Figs. 5.3, 5.7, and 5.11. Figures 5.4, 5.8, and 5.12 describe zoomed in excerpts in the graphs of the second time-derivatives. It can be concluded that the adaptive controller well operated in each case, independently of the asymmetries in the excitation.

5.1.2.2 Computations for Increasing Frequency and Amplitude

In the hope that a *dynamically more complex* trajectory may generate better $\{q, \dot{q}, \ddot{q}, u\}$ data set for learning a *nominal trajectory* was chosen so that both of its amplitude and frequency was linearly increased in time. In these estimations the parameter $c_4 = 2.0$ considerably improved the quality of the controlled motion. At first the *trajectory shift* -0.5 [m] was considered. In Fig. 5.13 relatively precise trajectory tracking can be observed the error of which is increasing in time at the end of the time session considered that is caused by the increase in the amplitude and frequency of the nominal trajectory. According to Fig. 5.14 it can be stated that only little extent of *adaptive deformation* was necessary the implicated that the *significant parameter combinations in the case of this trajectory* were well estimated. However, as it is testified by Fig. 5.15 the *individual parameters* were not appropriately estimated.

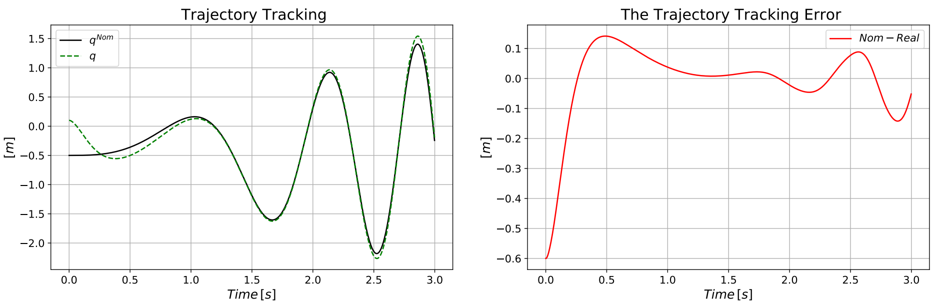

Figure 5.13: The trajectory tracking and its error for non-sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$

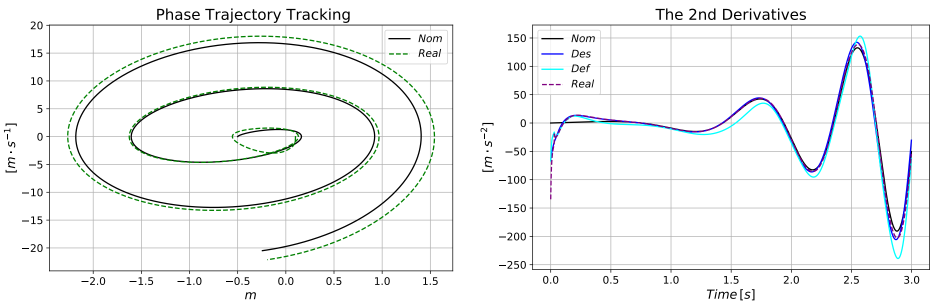

Figure 5.14: The phase trajectory tracking and the 2nd time-derivatives for non-sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$

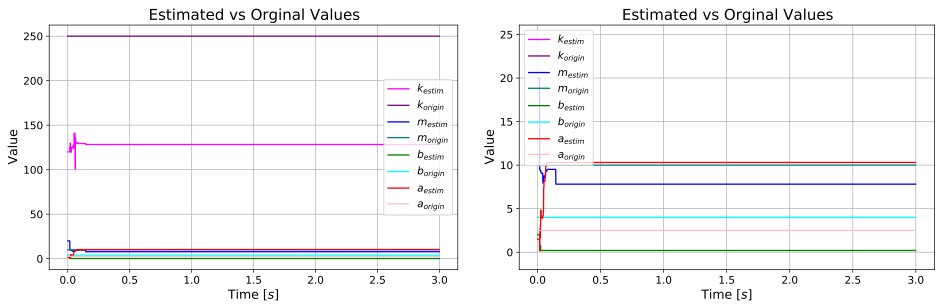

Figure 5.15: The estimated parameters for non-sinusoidal motion for trajectory shift of $-0.5\,[\mathrm{m}]$ (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

The figures belonging to the *trajectory shift* $0\,[\mathrm{m}]$ also reveal relatively precise trajectory tracking (Fig. 5.16) and relatively well identified *significant parameter combinations* (Fig. 5.17). Figure. 5.18 reveals that m was precisely estimated, b was found almost precisely, while the parameters as a and k had relatively little significance in this case because they were imprecisely estimated. This fact can be understood on the basis of the dynamic model: without trajectory shift relatively limited spring forces were generated, and the influence of the term $(q^2 - a^2)\,\dot{q}$ in (5.1a) was relatively insignificant, too. However, the estimation of m in the term $\frac{u}{m}$ was significant.

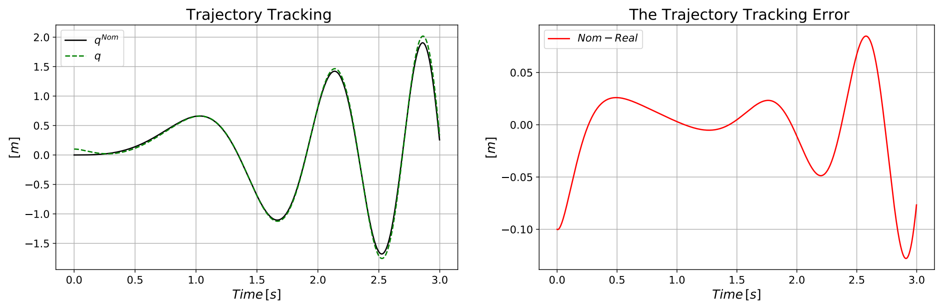

Figure 5.16: The trajectory tracking and its error for non-sinusoidal motion for trajectory shift of 0 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

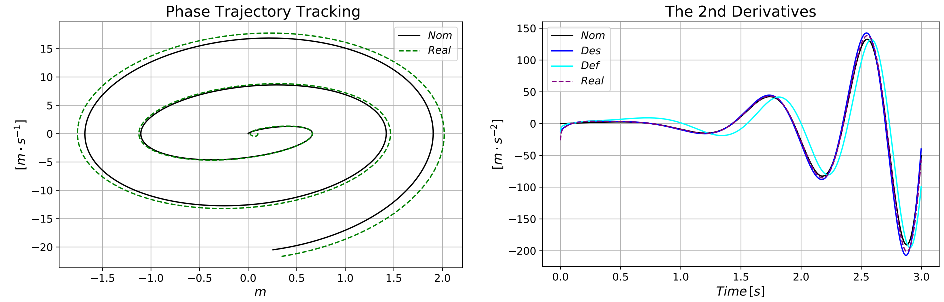

Figure 5.17: The phase trajectory tracking and the 2nd time-derivatives for non-sinusoidal motion for trajectory shift of 0 [m]

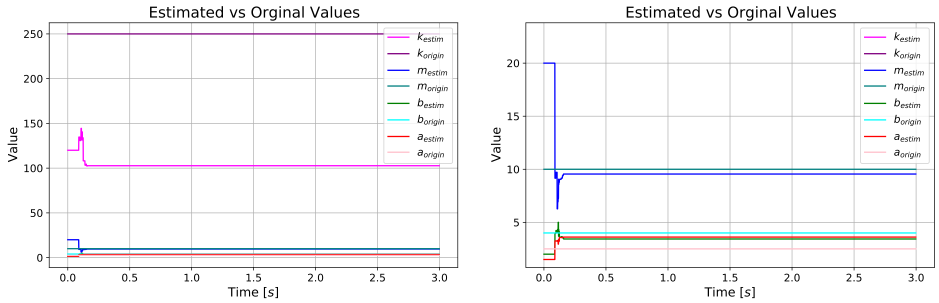

Figure 5.18: The estimated parameters for non-sinusoidal motion for trajectory shift of 0 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

Figures. 5.19–5.21 belonging to the *trajectory shift* 0.5 [m] also reveal that *m* was precisely estimated but the other ones were not very well found.

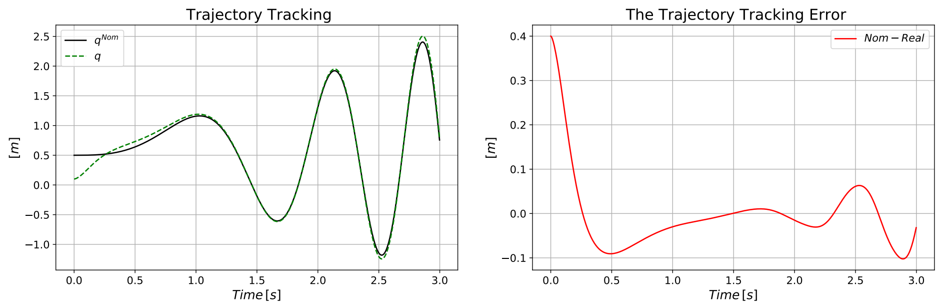

Figure 5.19: The trajectory tracking and its error for non-sinusoidal motion for trajectory shift of 0.5 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

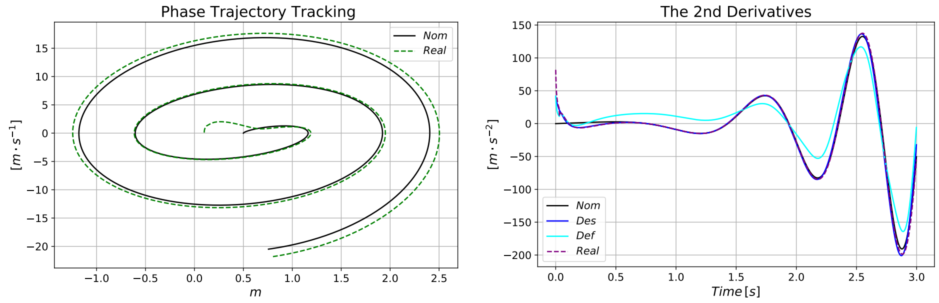

Figure 5.20: The phase trajectory tracking and the 2nd time-derivatives for non-sinusoidal motion for trajectory shift of 0.5 [m]

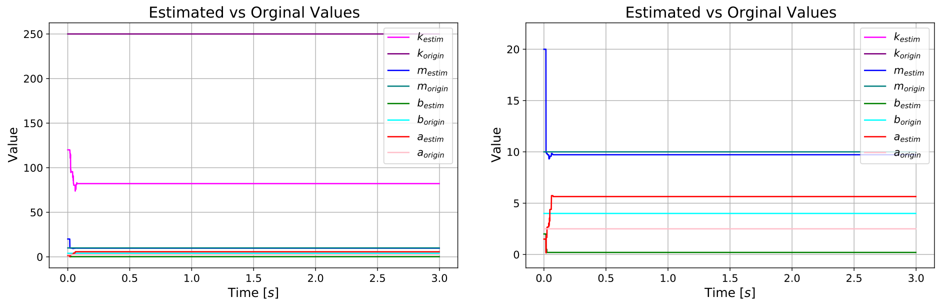

Figure 5.21: The estimated parameters for non-sinusoidal motion for trajectory shift of 0.5 [m] (In the RHS a zoomed in excerpt of the figure in the LHS is given.)

For the motion of increasing amplitude that takes more exhausting samples of the phase space than the sinusoidal motion with a fixed circular frequency, (see Figs. 5.14, 5.17, and 5.20) the following observations can be done:

- Following the decay of the initially great tracking error, due to the increase of the amplitude of the motion to be tracked, the tracking error started to slowly increase again, in all cases belonging to the shift parameters (Figs. 5.13, 5.16, and 5.19).

- In the second time-derivatives of the coordinate $q(t)$ in this case no jumps can be observed (Figs. 5.14, 5.17, and 5.20.) This statement is confirmed by the graphs of the estimated parameters (Figs. 5.15, 5.18, and 5.21) according to which the identification of the parameters happened in the early phase of the motion, and later no better solution was found by the PSO.

5.2 Application for a Realistic Robot Model

The main FPI-scheme is modified so that it allows the contribution of PSO. The main blocks remains to fit the control cycle while the addition are the PSO block and its cost function block. The PSO block takes the information from the realized coordinates q_1^{Real}, q_2^{Real}, q_3^{Real}, their first time derivatives $\dot{q}_1^{Real}$, $\dot{q}_2^{Real}$, $\dot{q}_3^{Real}$, and the control torque components Q_1, Q_2, Q_3 so that it can calculate their estimated second time derivatives $\ddot{q}_1^{Est}$, $\ddot{q}_2^{Est}$, $\ddot{q}_3^{Est}$. These estimated values will be evaluated and compared with the realized second derivatives $\ddot{q}_1^{Real}$, $\ddot{q}_2^{Real}$, $\ddot{q}_3^{Real}$ via the cost function defined as $Cost = \sum_i |\ddot{q}_i^{Real} - \ddot{q}_i^{Est}|$. Here, the usual process of PSO handles to search journey toward the global minimum in respect of the defined particles. The dynamics of the PSO-based parameter tuning is not coupled to that of the adaptive control, in contrast to the operation of the classic tuning methods. The iterative sequence of the FPI-based adaptive controller is formed within the loop as indicated in the chart by a red curve.

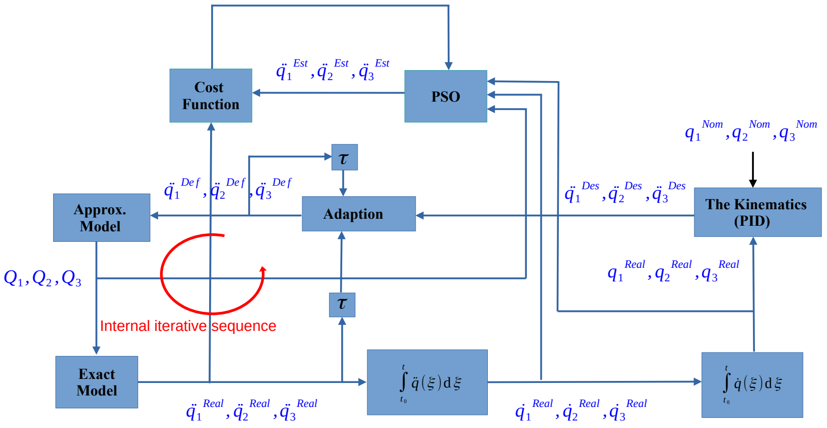
$\ddot{q}_1^{Est}, \ddot{q}_2^{Est}, \ddot{q}_3^{Est}$
Cost Function
PSO
$q_1^{Nom}, q_2^{Nom}, q_3^{Nom}$
τ
$\ddot{q}_1^{Def}, \ddot{q}_2^{Def}, \ddot{q}_3^{Def}$
Approx. Model
Adaption
$\ddot{q}_1^{Des}, \ddot{q}_2^{Des}, \ddot{q}_3^{Des}$
The Kinematics (PID)
Q_1, Q_2, Q_3
$q_1^{Real}, q_2^{Real}, q_3^{Real}$
τ
Internal iterative sequence
Exact Model
$\int_{t_0}^{t} \ddot{q}(\xi) d\xi$
$\int_{t_0}^{t} \dot{q}(\xi) d\xi$
$\ddot{q}_1^{Real}, \ddot{q}_2^{Real}, \ddot{q}_3^{Real}$
$\dot{q}_1^{Real}, \dot{q}_2^{Real}, \dot{q}_3^{Real}$

Figure 5.22: The PSO's operation flow chart.

The sequence of the adaptively deformed signals consists of the elements $\{\ddot{q}^{Def}(i); i \geq 1\}$ in which $\ddot{q}^{Def}(1) = \ddot{q}^{Des}(1)$, as can be seen in the feedback loop in Fig. 5.22, which also conveys information on the parameter identification process.

The essence of Algorithm 1 intuitively can be highlighted in the *symbolic picture* in Fig. 3.3.

Algorithm 1 The abstract rotation-based fixed point transformation algorithm

```
Require: $0 < \varepsilon \ll 1$, $\lambda_a \in ]0,1]$, $\ddot{q}^{Des}(t); \ddot{q}^{Real}(t); \ddot{q}^{Def}(t) \in \mathbb{R}^3$, $A; B; C \in \mathbb{R}^4$, $\ddot{q}_{max} \in \mathbb{R}^+$
Ensure: $\ddot{q}_{max} tanh(\frac{Transformed[1:3]}{\ddot{q}_{max}})$
    $A \leftarrow \left[\ddot{q}_1^{Des}(t) \quad \ddot{q}_2^{Des}(t) \quad \ddot{q}_3^{Des}(t) \quad \sqrt{|R_a^2 - A[1:3]^T A[1:3]|}\right]$
    $B \leftarrow \left[\ddot{q}_1^{Real}(t-1) \quad \ddot{q}_2^{Real}(t-1) \quad \ddot{q}_3^{Real}(t-1) \quad \sqrt{|R_a^2 - B[1:3]^T B[1:3]|}\right]$
    $C \leftarrow \left[\ddot{q}_1^{Def}(t-1) \quad \ddot{q}_2^{Def}(t-1) \quad \ddot{q}_3^{Def}(t-1) \quad \sqrt{|R_a^2 - C[1:3]^T C[1:3]|}\right]$
    $\|A\| = \|B\| = \|C\| = R_a$
    $A \perp B \leftarrow A - \frac{(B^T B)B}{R_a^2}$
    $\| A \perp B \| \leftarrow \sqrt{|A \perp B^T A \perp B|}$
    $e_\alpha \leftarrow \frac{A \perp B}{|\varepsilon + \|A \perp B\||}$;   $e_\beta \leftarrow \frac{B}{R_a}$;   $\varphi \leftarrow asin(\frac{\|A \perp B\|}{R_a})$
    $Gen \leftarrow e_\alpha e_\beta^T - e_\beta e_\alpha^T$
    $O \leftarrow I(4,4) + sin(\lambda_a \varphi) Gen + Gen^2 (1 - cos(\lambda_a \varphi))$        ▷ Rodrigues formula
    $Transformed \leftarrow OC$
```

With regard to the possible convergence of the adaptation process the general form of the equation of motion of the robots given in (5.3) can be considered.

$$H(q)\ddot{q} + h(q,\dot{q}) = Q \ , \tag{5.3}$$

If $\hat{H}(q)$ and $\hat{h}(q,\dot{q})$ are the *approximate model components*, and $H(q)$ and $h(q,\dot{q})$ denote the *exact ones*, by the use of the deformed signal $\ddot{q}^{Def}$ the realized $\ddot{q}^{Real}$ can be computed as follows:

$$Q = \hat{H}(q)\ddot{q}^{Def} + \hat{h}(q,\dot{q}) \ , \tag{5.4a}$$

$$\ddot{q}^{Real} = H^{-1}\hat{H}\ddot{q}^{Def} + H^{-1}(\hat{h} - h) \ . \tag{5.4b}$$

In [92], as the generalization of the concept of the single variable *monotonic increasing function* for multiple variable cases, the *approximately direction keeping* $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ function was defined in the following manner: if $\forall \Delta x$ the value $\Delta x^T \Delta f \equiv \Delta x^T \ (f(x+\Delta x) - f(x)) \approxeq \Delta x^T \frac{\partial f}{\partial x} \Delta x > 0$, then $f(x)$ is approximately direction keeping. Evidently, if someone wishes to achieve some desired Δf, in the case of such a function, he/she can *iteratively find* an appropriate input Δx. A car driver, who is an intelligent adaptive system, can learn driving a particular car if the steering wheel, the accelerator, and the brake pedals behave appropriately. In a similar manner, a fixed point iteration can be made convergent for such systems. If in (5.4b) the deformation is realized by the linear operator D as $\ddot{q}^{Def} = D\ddot{q}^{Des}$, it is concluded that

$$\frac{\partial \ddot{q}^{Real}}{\partial \ddot{q}^{Des}} = H^{-1}\hat{H}D \ , \tag{5.5}$$

that can be made approximately direction keeping by choosing $D := \mu \hat{H}^{-1} H$ with $\mu > 0$ leading to $\frac{\partial \ddot{q}^{Real}}{\partial \ddot{q}^{Des}} = \mu I$. That is, many appropriate adaptive deformations exist, and the fixed point iteration can find a solution to the problem. More sophisticated and more general considerations for convergence can be found in [120].

5.2.1 Model Dynamics

The simulations were made in Julia language with time resolution $\delta t = 10^{-3}\,[s]$ by using the 3-DOF robot system that can be realized by Equations (5.6)–(5.8) in which the following shortcuts were applied for simplifying the appearance as possible: $c_2 = \cos(q_2)$, $s_2 = \sin(q_2)$, $c_3 = \cos(q_3)$, $s_3 = \sin(q_3)$, $c_{23} = \cos(q_2 + q_3)$, $s_{23} = \sin(q_2 + q_3)$.

$$\begin{aligned} Q_1 = &\left(\Theta_1 + 0.25 m_2 L_2^2 c_2^2 + 0.25 m_3 L_3^2 c_{23}^2 + m_3 L_2^2 c_2^2 + 0.5 m_3 L_2 L_3 c_{23} c_2\right)\ddot{q}_1 \\ &+ \Bigg(-0.5 m_2 L_2^2 c_2 s_2 \dot{q}_2 - m_3 L_3^2 c_{23} s_{23} (\dot{q}_2 + \dot{q}_3)/2 - 2 m_3 L_2^2 c_2 s_2 \dot{q}_2 \\ &- 0.5 m_3 L_2 L_3 s_{23} c_2 (\dot{q}_2 + \dot{q}_3) - 0.5 m_3 L_2 L_3 c_{23} s_2 \dot{q}_2 \Bigg)\dot{q}_1 \quad , \end{aligned} \tag{5.6}$$

$$\begin{aligned} Q_2 = &\left(0.25 m_2 L_2^2 + 0.25 m_3 L_3^2 + m_3 L_2^2 + 0.5 m_3 L_3 L_2 c_3\right)\ddot{q}_2 + \left(0.25 m_3 L_3^2 + 0.25 m_3 L_3 L_2 c_3\right)\ddot{q}_3 \\ &- 0.5 m_3 L_3 L_2 s_3 \dot{q}_3 \dot{q}_2 - 0.25 m_3 L_3 L_2 s_3 \dot{q}_3^2 + \Bigg(0.25 m_2 L_2^2 c_2 s_2 + 0.25 m_3 L_3^2 c_{23} s_{23} + m_3 L_2^2 c_2 s_2 \\ &+ 0.25 m_3 L_2 L_3 s_{23} c_2 + 0.25 m_3 L_2 L_3 c_{23} s_2 \Bigg)\dot{q}_1^2 + 0.5 m_2 L_2 g c_2 + m_3 g L_2 c_2 + 0.5 m_3 L_3 g c_{23} \quad , \end{aligned} \tag{5.7}$$

$$\begin{aligned} Q_3 = &\left(0.25 m_3 L_3^2 + 0.25 m_3 L_3 L_2 c_3\right)\ddot{q}_2 + \left(0.25 m_3 L_3^2\right)\ddot{q}_3 + 0.25\left(m_3 L_3^2 c_{23} s_{23} + m_3 L_3 L_2 s_{23} c_2\right)\dot{q}_1^2 \\ &+ 0.25 m_3 L_3 L_2 s_3 \dot{q}_2^2 + 0.5 m_3 g L_3 c_{23} \quad . \end{aligned} \tag{5.8}$$

Table 5.1 contains the parameters of the exact model to realize the original system behavior and the approximate model parameters used by the adaptive controller.

Table 5.1: The robot model parameters in Equations (5.6)–(5.8).

Parameter	Exact Model	Approximate Model
Θ_1 [kg · m^2] 1st link's inertia moment	50.0	55.0
m_2 [kg] 2nd link mass	10.0	8.0
m_3 [kg] 3rd link mass	20.0	18.0
L_2 [m] 2nd link length	2.0	2.0
L_3 [m] 3rd link length	1.0	1.0
g [m · s^{-2}] gravitational accel.	9.81	9.81

5.2.2 Implementation of the PSO Strategy

For estimation of the robot model, five parameters ($n = 5$) were chosen: Θ_1, L_2, L_3, m_2, and m_3. These parameters were placed in a row as input. For the effective minimum value search, the sixth place was reserved for the cost function, which was computed by the absolute value of the difference between the estimated $\hat{\ddot{q}}$ and the realized (observed) $\ddot{q}^{Real}$ values.

For initializing the PSO particles, a non-empty grid of points was set (i.e., it consisted of a set of random populations with *Initial Values* $\{[\Theta_{1_{ini}}, L_{2_{ini}}, L_{3_{ini}}, m_{2_{ini}}, m_{3_{ini}}]\}$) so that the evaluated particles could move accordingly. The number of particles was set to 32 on the basis of the idea as follows. It was assumed that each parameter p_i might have an estimated value δp_i, and might have a lower and an upper estimation $\delta p_{il} < \delta p_i < \delta p_{iu}$. While the δp_i values were fixed numbers, the lower and upper estimations were set randomly. For these estimations in dimension n the necessary minimal number of parameters is 2^n. For $n = 3$ see Fig. 5.23.

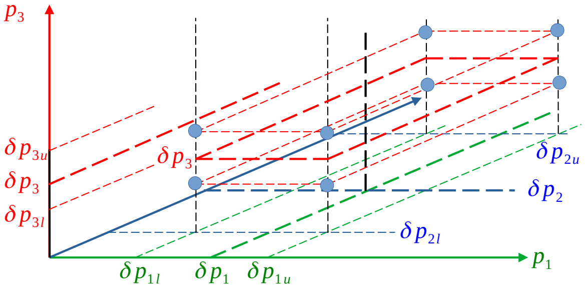

Figure 5.23: Idea for selecting the PSO parameters.

The equation of motion of the particles had the same form as in (5.2).

The flowchart in Fig. 5.22 explains the signals directly fed from FPI-based control $q^{Real}, \dot{q}^{Real}$, and Q that are used to calculate the estimated acceleration for each robot link as $\ddot{q}_1^{Est}, \ddot{q}_2^{Est}, \ddot{q}_3^{Est}$. These estimated values are compared to the real second coordinate time derivatives that are obtained from the FPI control cycle by using the cost function evaluated by the PSO algorithm.

5.2.3 Simulation Results

The simulation work consists of two parts: the first realizes the parameter identification by PSO, while the second tests the CTC controller using the identified parameters. In the second part, the operation of the adaptive controller with the identified parameters is considered.

I. Identifying Parameters by PSO

According to the considerations with regard to the significance of the well balanced teaching set in parameter learning, the FPI-based adaptive controller was used for realizing or at least well approximating a nominal trajectory that was invented for teaching purposes. Figure. 5.24 shows the responses of trajectory tracking properties for the three robot links. As it can be seen in Figs. 5.25–5.27, which describe the phase trajectory of the motion used for teaching, several boxes in the phase space of each link are visited. Though this is far from the exactly even distribution over the cells, it seems to be more or less well balanced. Figure. 5.28 shows how fast the convergence of the PSO algorithm is. Essentially each parameter was *almost perfectly* identified.

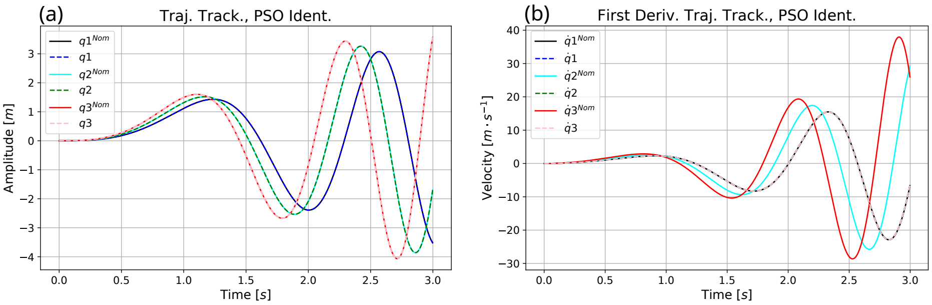

Figure 5.24: Trajectory tracking (**a**) and the first time derivatives (**b**) for the three robot links in the parameter identification process.

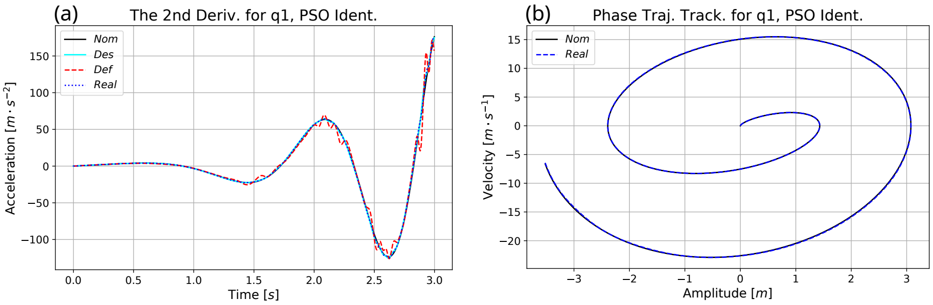

Figure 5.25: The second time derivatives (**a**) and phase trajectory tracking (**b**) for q_1.

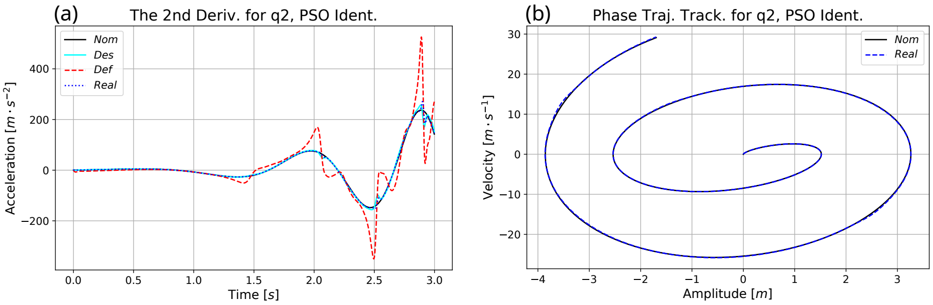

Figure 5.26: The second time derivatives (**a**) and phase trajectory tracking (**b**) for q_2.

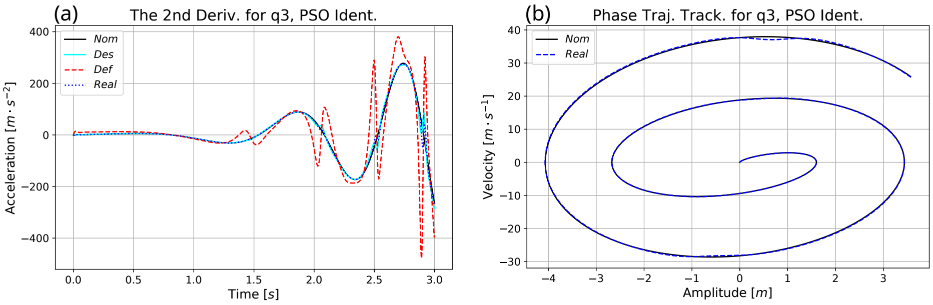

Figure 5.27: The second time derivatives (**a**) and phase trajectory tracking (**b**) for q_3.

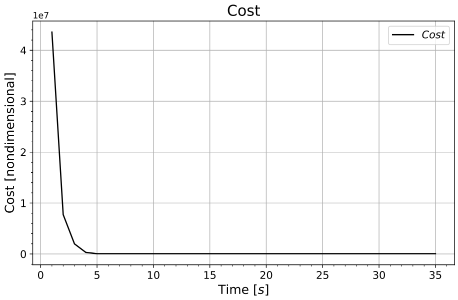

Figure 5.28: The cost function of the best estimation of the model during PSO teaching iteration.

The accuracy of each estimated value for each targeted parameter is shown in Figure. 5.29. For revealing the estimation accuracy in the five-dimensional parameter space,

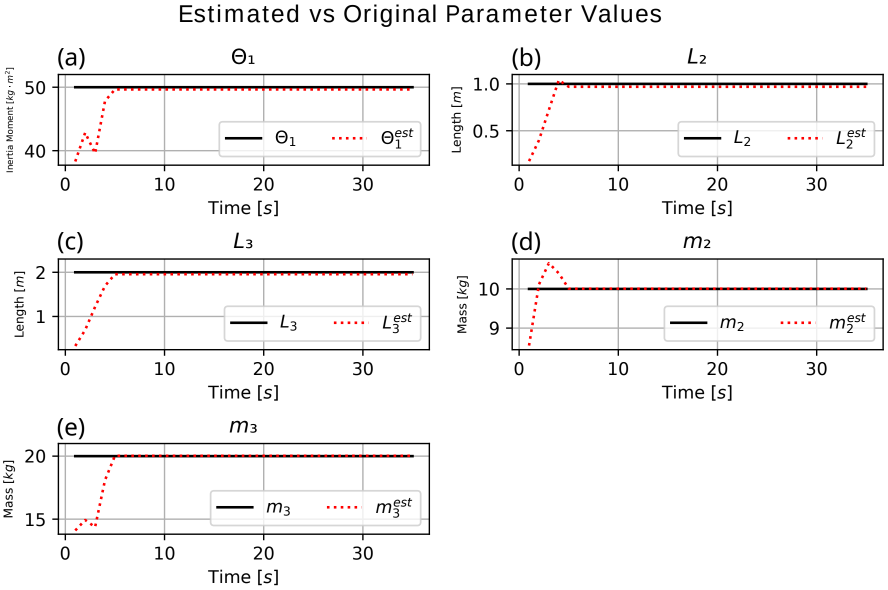

Figure 5.29: The estimated vs original parameters during teaching iteration (**a**) for Θ_1 (**b**) for L_2 (**c**) for L_3 (**d**) for m_2 (**e**) for m_3.

II. Operation of the Non-Adaptive CTC vs. Adaptive CTC for the Identified Parameters

After identifying the appropriate parameters by PSO, a natural option would be using the common non-adaptive CTC in the possession of the identified model. However, the computations revealed that the small errors in the identified parameters are not completely insignificant. It may be advantageous to maintain the adaptive controller even in the case in which the identified dynamic parameters are used. For this reason, two versions of the CTC controller were applied: one without any adaption technique, while the other by activating the FPI-based adaptive mechanism. It can be expected that the effect of adaptivity will be more significant in the identification phase when a very imprecise model will be used for tracking a prescribed trajectory. It can be expected that its significance will not be so great when a more precise model will be in use.

In fact, the trajectory tracking error in the PSO-based identifying process was between -0.01[m] and 0.0075[m] whereas it increased in non-adaptive CTC within the range $[-0.02, 0.02]$ [m]. The better improvement was in the adaptive CTC within the range $[-0.008, 0.004]$ [m]. The same holds for the first derivative trajectory tracking error. It increased from $[-0.75, 0.75][\mathrm{m}\cdot\mathrm{s}^{-1}]$ in the original identifying process to $[-1.0, 1.0][\mathrm{m}\cdot\mathrm{s}^{-1}]$ in non-adaptive CTC. The error decreased to $[-0.35, 0.30][\mathrm{m}\cdot\mathrm{s}^{-1}]$ in adaptive CTC. The comparison can be seen in Figs. 5.30–5.32.

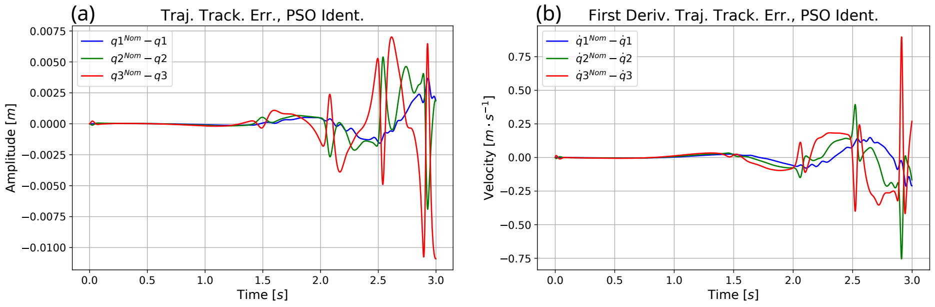

Figure 5.30: Trajectory tracking error (**a**) and its first time derivative (**b**) during the parameter identification process.

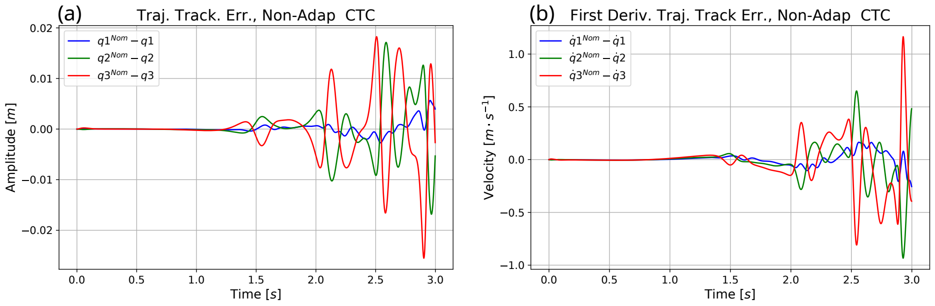

Figure 5.31: Trajectory tracking error (**a**) and its first time derivative (**b**) for the non-adaptive CTC controller using the identified parameters.

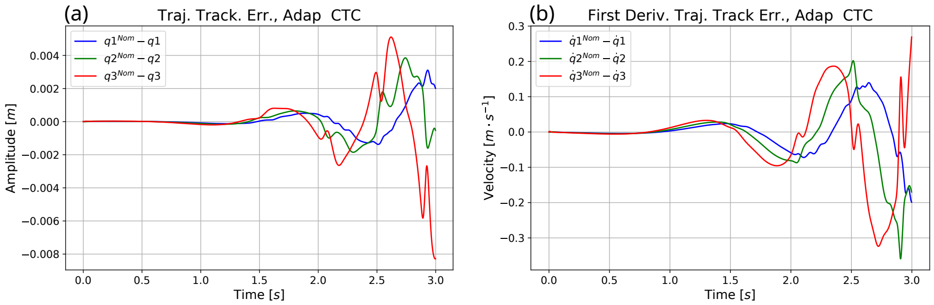

Figure 5.32: Trajectory tracking error (**a**) and its first time derivative (**b**) for the adaptive CTC controller using the identified parameters.

In general, it can be expected that in the case of an imprecise model the non-adaptive controller has to apply significant PID error correction terms to the second time derivatives of the nominal coordinates; therefore, in this case, quite considerable differences can be expected between the values of $\ddot{q}_1^{Nom}(t)$, $\ddot{q}_2^{Nom}(t)$, $\ddot{q}_3^{Nom}(t)$, and $\ddot{q}_1^{Des}(t)$, $\ddot{q}_2^{Des}(t)$, $\ddot{q}_3^{Des}(t)$: the significant additions in the desired terms have the role of making the necessary corrections in the non-adaptive controller. For a good operation, the Realized values should well track these Desired ones, but this cannot be well realized in the non-adaptive controller. The limited applicability of the non-adaptive controller consists in this fact. However, when the adaptation mechanism is in use, it is expected that the Realized value better approximates the Desired one, consequently the necessary PID-type corrections in the Desired term continuously decrease in time, therefore both the Desired and the Realized values converge to the Nominal ones, while the Deformed ones can increase accordingly. (As the final results, the precision of trajectory tracking and that of the first time derivatives of the generalized coordinates can be compared by the use of Figs. 5.30–5.32. It can be seen that the non-adaptive controller that uses the identified parameters has the greatest error, the second greatest error belongs to the adaptive CTC controller using the original data, and finally, the most precise tracking was achieved by the adaptive controller using the identified model parameters.)

The above-mentioned effects can be well observed in the simulation results when the not completely precise identified parameters were used in the non-adaptive and adaptive versions of the CTC controller. This effect can be well tracked in the zoomed-in excerpts of Figs. 5.33 and 5.34 for the coordinate q_1.

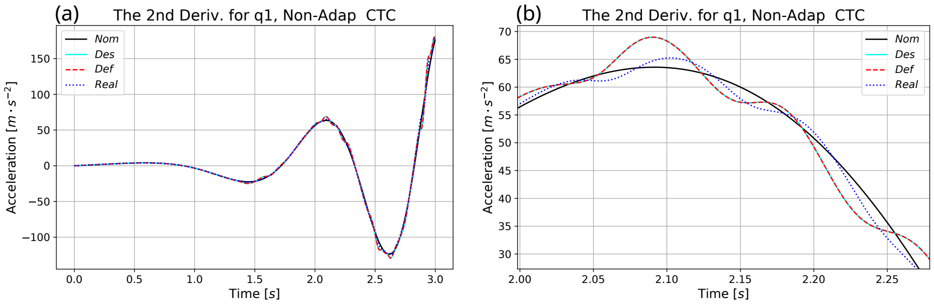

Figure 5.33: Second time derivatives for q_1 (**a**) and the zoomed-in excerpts (**b**) for the non-adaptive CTC controller using the identified parameters.

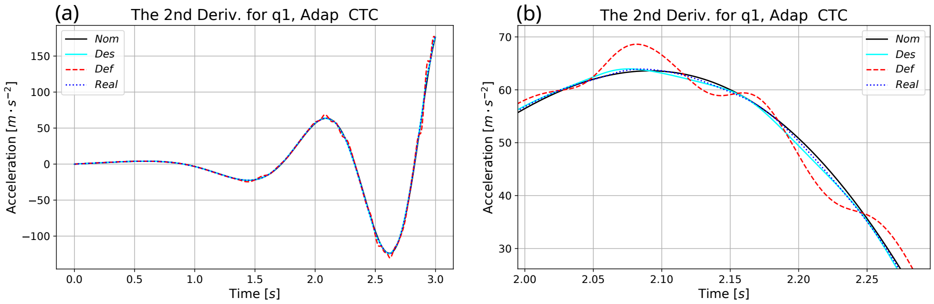

Figure 5.34: Second time derivatives for q_1 (**a**) and the zoomed-in excerpts (**b**) for the adaptive CTC controller using the identified parameters.

Similar effects can be well identified in Figs. 5.35 and 5.36 for the coordinate q_2, and in Figs. 5.37 and 5.38, for the coordinate q_3, too.

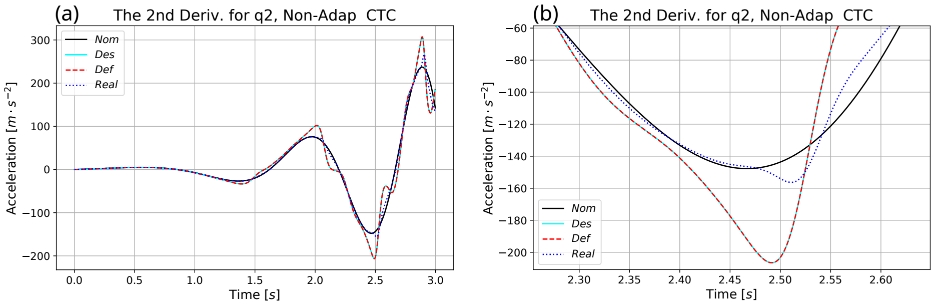

Figure 5.35: Second time derivatives for q_2 (**a**) and the zoomed-in excerpts (**b**) for the non-adaptive CTC controller using the identified parameters.

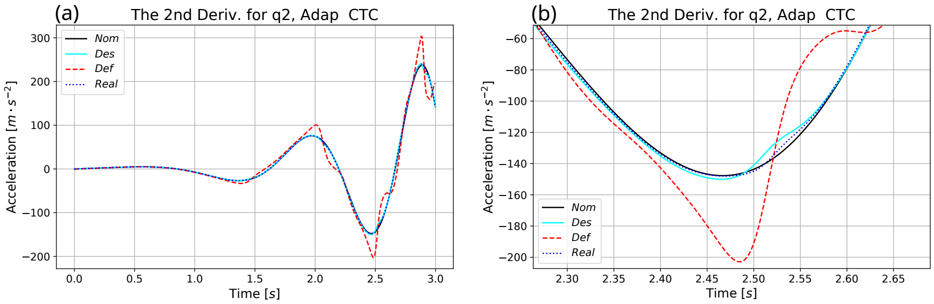

Figure 5.36: Second time derivatives for q_2 (**a**) and the zoomed-in excerpts (**b**) for the adaptive CTC controller using the identified parameters.

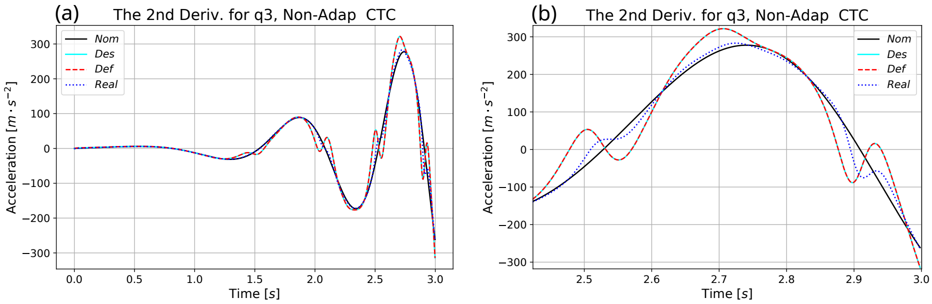

Figure 5.37: Second time derivatives for q_3 (**a**) and the zoomed-in excerpts (**b**) for the non-adaptive CTC controller using the identified parameters.

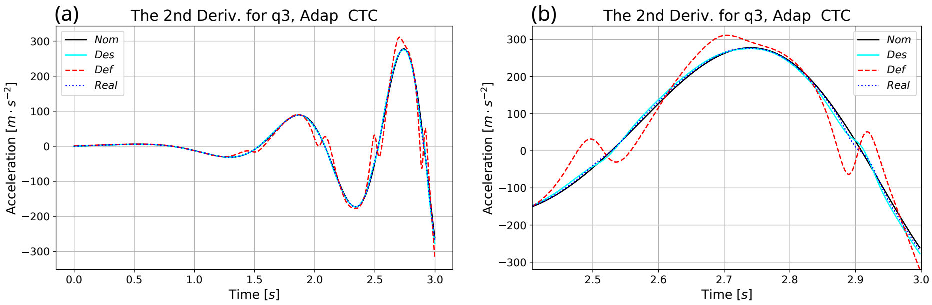

Figure 5.38: Second time derivatives for q_3 (**a**) and the zoomed-in excerpts (**b**) for the adaptive CTC controller using the identified parameters.

Though for the very imprecise initial model no non-adaptive simulations were done, in Figs. 5.39–5.41, similar observations can be done for the coordinates q_1, q_2, and q_3, respectively. The adaptive controller brought closer to each other the $\ddot{q}^{Nom}$, $\ddot{q}^{Des}$, and $\ddot{q}^{Real}$ values by applying a considerable extent of adaptive deformations.

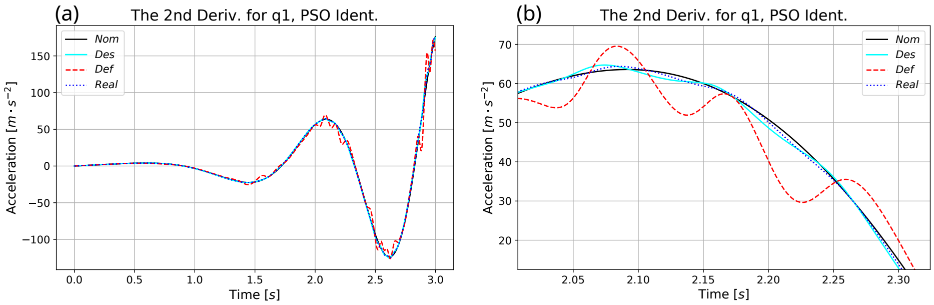

Figure 5.39: Second time derivatives for q_1 (**a**) and the zoomed-in excerpt (**b**) during the parameter identification process.

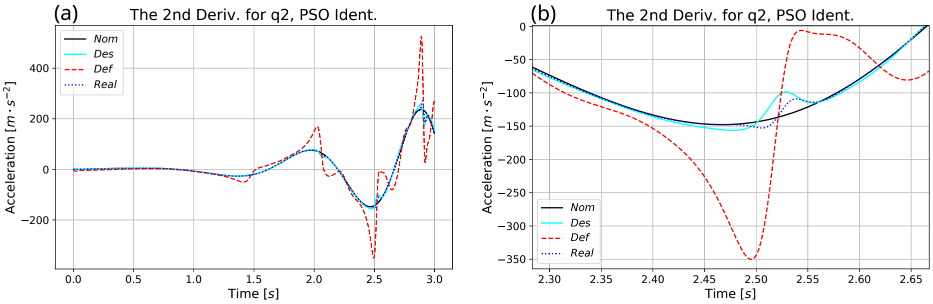

Figure 5.40: Second time derivatives for q_2 (**a**) and the zoomed-in excerpt (**b**) during the parameter identification process.

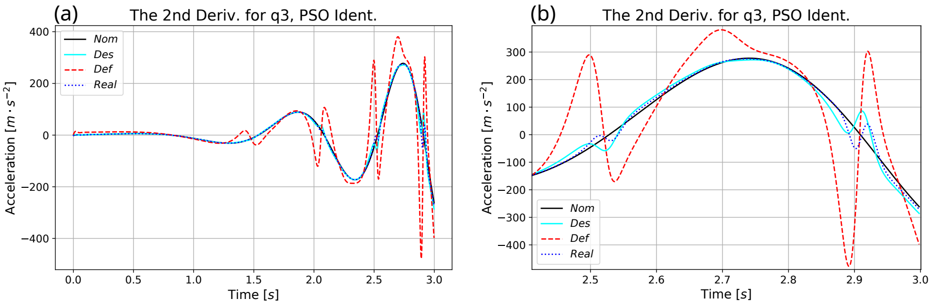

Figure 5.41: Second time derivatives for q_3 (**a**) and the zoomed-in excerpt (**b**) during the parameter identification process.

A possible measure of extent of adaptive deformation is the *angle of abstract rotation* in this special adaptive controller. The figure. 5.42 underpins the fact that: the use of the very imprecise model during the parameter identification process needed a much more drastic adaptive deformation than the adaptive use of the identified parameters.

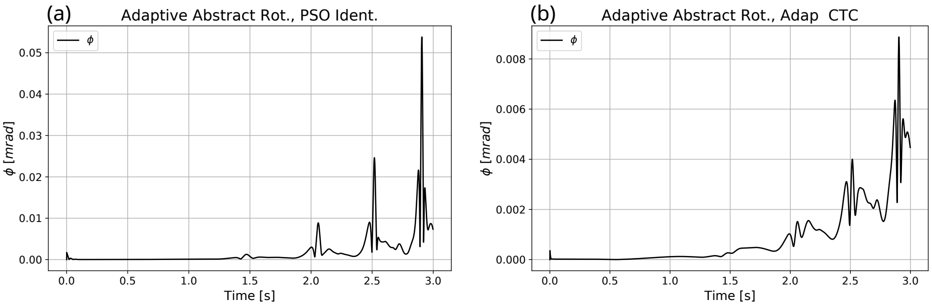

Figure 5.42: The angle of adaptive abstract rotation during the parameter identification process (**a**), and for the adaptive CTC controller using the identified parameters (**b**).

5.3 Thesis Statement IV

I have elaborated a cooperation between the Fixed Point Iteration-based adaptive

control and a system parameter identification process using Particle Swarm Optimization. The method allowed online and offline modes. In the offline mode the adaptive control and the model identification tasks were completely decoupled, and the quality of trajectory tracking was independent of the actual phase of identification, and was precise from the beginning of the controller's operation.

I have shown that due to balancing problems in the online case no asymptotic convergence of the identified model can be expected. The fact that the use of the online mode is not compulsory in the novel approach is an advantage over the traditional ones that cannot avoid the application of the not well identified model from the beginning of the control.

***Related own publications:* [A. 10] and [A. 11].**

Chapter 6

Conclusions

In my Dissertation I brought about and investigated the integration of two subject areas in control technology: the optimal and the adaptive controllers. After realizing that both subject areas have their traditional mathematical backgrounds, namely the functional minimization in optimal control, and the use of Lyapunov functions tailored to the particular problem under considerations in adaptive control, that makes their integration mathematically very complicated and difficult, I realized that in a particular slot this integration is not difficult. It was the fixed point iteration-based adaptive control approach. I have shown that its simple mathematical structure offers plausible possibilities for combination and integration with the heuristic optimal controllers in which a great variety of cost functions and constraints can be applied. To utilize the so created freedom I have evaded the traditional optimization approach in which the dynamic model is used as a set of constraint equations. I reduced the number of constraint equations to one, and later completely evaded the use of constraint terms by building the dynamic model into the cost function at the price of little redundancies in the computations. I have shown that these variants were able to cooperate with the adaptive technique I have applied, while allowed simplifications in finding sub-optimal solutions in the case of various complicated cost terms in Receding Horizon controllers. Besides control issues I suggested the application of this method in the solution of differential inverse kinematic task of redundant robot arms to efficiently tame the solution in the vicinity of kinematic singularities.

Since the adaptive method applied was expected to be noise sensitive, I suggested its combination with various simple noise filtering techniques.

I also improved the adaptive method that originally suffered from some deficiency related to actuator saturation issues.

Finally, I have elaborated the combination and elaboration of the adaptive control applied with parameter estimation task that have significance independently of the need of precise control. I have realized that by the use of the fixed point iteration-based approach the control and identification tasks can be optionally separated from each other, and arrived at the conclusion that due to balancing problems the precision of the estimation has limitations, and in general, it is expedient to use the improved model with some adaptive technique. I have shown that if the adaptation has to make only small corrections, its stability range increases and becomes much more reliable than the original approach using a very rough initial dynamic system model.

Chapter 7

Future Research

Taking into account that the basic adaptation mechanism of the fixed point iteration-based solution has a typical modular structure in which the *kinematic* and *dynamic* issues are separated from each other, it offers various further development possibilities as follows:

i) In the kinematic block that defines how the tracking error required to converge to zero or kept near zero various ideas can be applied. In my Thesis I mainly used PID-type requirements, but any ideas related to the use of certain results of fractional order calculus or fractional order derivatives-inspired approaches can be incorporated in this block;

ii) Though the main idea of FPI-based approach was considering purely kinematic requirements, the FPI-based Model Reference Adaptive Control variant of the method can be combined with model- and Lyapunov function-based approaches as backstepping controllers in which the kinematic and dynamic issues are not strictly separated from each other but various Lyapunov functions can be conveniently used;

iii) The method may be applicable for making a large number of simulations how the variables of the internal compartment components can vary in time when these variables cannot be measured, and we are in lack of satisfactory information even for estimating them (e.g., in life sciences).

iv) The main advantage of the method from the point of view of implementation is that instead using complicated analytical formulae in the approximate models, often quite simple estimations can be used for the purpose of control so that the deficiencies of the model can be evaded by a very simple adaptive mechanism. The little computational demand of the simple model (e.g., an affine model) can substitute the computation of very complicated Lie derivatives. If the simple model can be used by an embedded system, the computational force of humble hardware systems can be satisfactory from practical point of view. The main limitation if the FPI-based method, i.e., that during one digital control step only one step of the adaptive iteration can be done generally can be amended by decreasing the duration of the control cycle e.g., by the use of stronger hardware. This simple observation explains that the huge computational need of the Dynamic Programming invented in the fifties of the past century made it applicable for controlling slow chemical processes with hardware solutions of humble computational capacities, while for controlling fast robot motion, even in the technological background of the late nineties was satisfactory only for the realization of the simple CTC controller. In the parallel research by PhD student Awudu Atinga the problem of adaptive control of underactuated by the use of fixed point iteration is considered in details with numerical simulations for an example that does not increase the relative order of the control task and for an other one in which for the control of a second order system is solved by a relative order 3 controller [121].

v) For implementation issues of the fixed point iteration-based method for real life applications, in a parallel research, Bence Varga PhD student keeps working on experimental investigations. He developed an experimental setup and obtained measurement values that presently are under review for publication. His system is made of typical cheap components that are appropriate for playing with them in the university research. A 12 V DC motor type FIT0185 with an inbuilt encoder and planetary gearbox with 131:1 reduction ratio. This electric motor is driven by a BTS7960-M dual half bridge motor drive. The output shaft is connected to a spring through a special coupling which provides a position-varying external load not modeled by the controller. Its effect is compensated by fixed point iteration-based adaptive CTC controller. The electronics consists of Arduino DUE Board with Atmel SAM3X8E ARM Cortex-M3 32-bit processor. The adaptive deformation is applied for the PWM (Pulse Width Modulation) output of the controller.

Appendix A

Regarding to section 2.1:
The estimation of the gradient and making the corrective steps:

```
part_time1=@elapsed ∂G=gradient("G",x_act)
part_time2=@elapsed x_act=x_act-(a_{1}*G_act/(∂G'*∂G))*∂G
Time[counter]=part_time1+part_time2
```

In the Reduced Gradient phase the main steps were as follows: filling in the columns of the matrix that subsequently is orthogonalized, taking out the last column for the reduced gradient, making the corrective steps, and calculation of the norm of the reduced gradient that is used for stopping the algorithm.

```
part_time1=@elapsed Xact[:,1:1]=gradient("g_{1}",x_act)
part_time2=@elapsed Xact[:,2:2]=gradient("g_{2}",x_act)
part_time3=@elapsed Xact[:,3:3]=-gradient("f",x_act)
part_time4=@elapsed reduced_matrix=GrSchm(Xact)
part_time5=@elapsed reduced=reduced_matrix[:,3:3]
part_time6=@elapsed x_act=x_act+a_{2}*reduced
part_time7=@elapsed red_grad_norm=norm(reduced)
```

In this case the Gram-Schmidt algorithm was realized by the code:

```
function GrSchm(X)
Xv=X # at the beginning
if norm(Xv[:,1:1])>varepsilon
L_{21}=(Xv[:,1:1]'*X[:,2:2])[1]/((Xv[:,1:1]'*Xv[:,1:1])[1])
L_{31}=(Xv[:,1:1]'*X[:,3:3])[1]/((Xv[:,1:1]'*Xv[:,1:1])[1])
else
L_{21}=0.0
L_{31}=0.0
end
Xv[:,2:2]=X[:,2:2]-L_{21}*Xv[:,1:1]
if norm(Xv[:,2:2])>varepsilon
L_{32}=(Xv[:,2:2]'*X[:,3:3])[1]/((Xv[:,2:2]'*Xv[:,2:2])[1])
else
L_{32}=0.0
end
Xv[:,3:3]=X[:,3:3]-L_{31}*Xv[:,1:1]-L_{32}*Xv[:,2:2]
return Xv
end
```

in which the symbols L_{21}, L_{31}, L_{32} determine the factors of reduction. In the second case this function was simpler because the reduction happened only by a single column as

```
function GrSchm(X)
```

```
Xv=X # at the beginning
if norm(Xv[:,1:1])>varepsilon
L21=(Xv[:,1:1]’*X[:,2:2])[1]/((Xv[:,1:1]’*Xv[:,1:1])[1])
else
L21=0.0
end
Xv[:,2:2]=X[:,2:2]-L21*Xv[:,1:1]
return Xv
end
```

Own Publications

Publications Strictly Related to the Dissertation

[A. 1] H. Issa and J. K. Tar, “Speeding up the reduced gradient method for constrained optimization,” in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI).* IEEE, 2021, pp. 485–490.

[A. 2] H. Issa, B. Varga, and J. K. Tar, “A receding horizon-type solution of the inverse kinematic task of redundant robots,” in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI).* IEEE, 2021, pp. 231–236.

[A. 3] H. Issa and J. K. Tar, “Tackling actuator saturation in fixed point iteration-based adaptive control,” in *2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI).* IEEE, 2020, pp. 221–226.

[A. 4] H. Issa, B. Varga, and J. K. Tar, “Accelerated reduced gradient algorithm for solving the inverse kinematic task of redundant open kinematic chains,” in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI).* IEEE, 2021, pp. 387–392.

[A. 5] H. Issa, H. Khan, and J. K. Tar, “Suboptimal adaptive receding horizon control using simplified nonlinear programming,” in *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES).* IEEE, 2021, pp. 221–228.

[A. 6] H. Issa and J. K. Tar, “Preliminary design of a receding horizon controller supported by adaptive feedback,” *Electronics*, vol. 11, no. 8, p. 1243, 2022.

[A. 7] H. Issa and J. K. Tar, “Noise sensitivity reduction of the fixed point iteration-based adaptive control,” in *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY).* IEEE, 2021, pp. 171–176.

[A. 8] H. Issa and J. K. Tar, “Comparison of Various Noise Filtering Techniques in Strongly Nonlinear Adaptive Control,” in *2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI).* IEEE, 2022, pp. 347–352.

[A. 9] H. Issa, M. Al-Bkree, and J. K. Tar, “On Certain Noise Filtering Techniques in Fixed Point Iteration-based Adaptive Control,” *SYSTEM THEORY, CONTROL AND COMPUT-*

ING JOURNAL, vol. 2, no. 2, pp. 9–16, 2022.

[A. 10] H. Issa and J. K. Tar, “On the Limitations of PSO in Cooperation with FPI-based Adaptive Control for Nonlinear Systems,” in *2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES).* IEEE, 2022, pp. 181–186.

[A. 11] H. Issa and J. K. Tar, “Improvement of an Adaptive Robot Control by Particle Swarm Optimization-Based Model Identification,” *Mathematics*, vol. 10, no. 19, p. 3609, 2022.

Other Publications

[A. 12] H. Khan, H. Issa, and J. K. Tar, “Comparison of the operation of fixed point iteration-based adaptive and robust vs/sm-type solutions for controlling two coupled fluid tanks,” in *2020 IEEE 20th International Symposium on Computational Intelligence and Informatics (CINTI).* IEEE, 2020, pp. 29–34.

[A. 13] H. Khan, H. Issa and J. K. Tar, “Improved simple noise filtering for fixed point iteration-based adaptive controllers,” in *2020 IEEE 20th International Symposium on Computational Intelligence and Informatics (CINTI).* IEEE, 2020, pp. 77–82.

[A. 14] H. Khan, H. Issa and J. K. Tar, “Application of the robust fixed point iteration method in control of the level of twin tanks liquid,” *Computation*, vol. 8, no. 4, p. 96, 2020.

[A. 15] B. Varga, H. Issa, R. Horváth, and J. K. Tar, “Sub-optimal solution of the inverse kinematic task of redundant robots without using Lagrange multipliers,” *SYSTEM THEORY, CONTROL AND COMPUTING JOURNAL*, vol. 1, no. 2, pp. 40–48, 2021.

[A. 16] B. Varga, H. Issa, R. Horváth, and J. K. Tar, “Accelerated reduced gradient algorithm with constraint relaxation in differential inverse kinematics,” *SYSTEM THEORY, CONTROL AND COMPUTING JOURNAL*, vol. 1, no. 2, pp. 21–32, 2021.

References

[1] N. Moldoványi, *Model Predictive Control of Crystallisers.* PhD Thesis, Department of Process Engineering, University of Pannonia, Veszprém, Hungary, 2012.

[2] R. Bellman, "Dynamic programming and a new formalism in the calculus of variations," *Proc. Natl. Acad. Sci.*, vol. 40, no. 4, pp. 231–235, 1954.

[3] R. Bellman, *Dynamic Programming.* Princeton Univ. Press, Princeton, N. J., 1957.

[4] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[5] V. Arnold, *Mathematical Methods of Classical Mechanics.* Springer - Verlag, 1989.

[6] J. Lagrange, J. Binet, and J. Garnier, *Mécanique analytique (Eds. J.P.M. Binet and J.G. Garnier).* Ve Courcier, Paris, 1811.

[7] J. Riccati, "Animadversiones in aequationes differentiales secundi gradus (observations regarding differential equations of the second order)," *Actorum Eruditorum, quae Lipsiae publicantur, Supplementa,*, vol. 8, pp. 66–73, 1724.

[8] E. Haynsworth, "On the Schur complement," *Basel Mathematical Notes*, vol. BMN 20, p. 17, 1968.

[9] A. Laub, *A Schur Method for Solving Algebraic Riccati Equations (LIDS-P 859 Research Report).* MIT Libraries, Document Services, After 1979.

[10] R. Kalman, "Contribution to the theory of optimal control," *Boletin Sociedad Matematica Mexicana*, vol. 5, no. 1, pp. 102–119, 1960.

[11] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and internal parameters of the PUMA 560 arm," *Proc. IEEE Conf. On Robotics and Automation 1986*, pp. 510–518, 1986.

[12] P. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," *Proc. IEEE Conf. Robotics and Automation, 1994*, pp. 1608–1613, 1994.

[13] B. Armstrong-Helouvry, "Stick slip and control in low-speed motion," *IEEE Trans. on Automatic Control*, vol. 38, no. 10, pp. 1483–1496, 1990.

[14] B. Armstrong-Helouvry, *Control of machines with friction*. Kluwer Academic Publishers, 1991.

[15] B. Armstrong-Helouvry, P. Dupont, and C. Canudas de Wit, “A survey of models, analysis tools and compensation methods for the control of machines with friction,” *Automatica*, vol. 30, no. 7, p. 1083, 1994.

[16] L. Márton and B. Lantos, “Identification and model-based compensation of striebeck friction,” *Acta Polytechnica Hungarica*, vol. 3, no. 3, pp. 45–58, 2006.

[17] K. Åström and C. Canudas de Wit, “Revisiting the LuGre friction model,” *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 101–114, 2008.

[18] S. Bennett, “Nicholas Minorsky and the automatic steering of ships,” *IEEE Control Systems Magazine*, vol. 4, no. 4, pp. 10–15, 1984.

[19] F. Mohd Zaihidee, S. Mekhilef, and M. Mubin, “Robust speed control of pmsm using sliding mode control (smc)—a review,” *Energies*, vol. 12, no. 9, p. 1669, 2019.

[20] S. Sagara and R. Ambar, “Performance comparison of control methods using a dual-arm underwater robot-computed torque based control and resolved acceleration control for uvms,” in *2020 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1094–1099, IEEE, 2020.

[21] M. Hamandi, M. Tognon, and A. Franchi, “Direct acceleration feedback control of quadrotor aerial vehicles,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5335–5341, IEEE, 2020.

[22] S. Yuan, M. Lv, S. Baldi, and L. Zhang, “Lyapunov-equation-based stability analysis for switched linear systems and its application to switched adaptive control,” *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2250–2256, 2020.

[23] K. M. Dogan, T. Yucelen, W. M. Haddad, and J. A. Muse, “Improving transient performance of discrete-time model reference adaptive control architectures,” *International Journal of Adaptive Control and Signal Processing*, vol. 34, no. 7, pp. 901–918, 2020.

[24] I. Sekaj and V. Veselý, “Robust output feedback controller design: Genetic Algorithm approach,” *IMA J Math Control Info*, vol. 22, no. 3, pp. 257–265, 2005.

[25] J. Chen and W.-D. Chang, “Feedback linearization control of a two-link robot using a Multi-Crossover Genetic Algorithm,” *Expert Systems with Applications*, vol. 2, no. 2 Part 2, pp. 4154–4159, 2009.

[26] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[27] B. Varga, J. Tar, and R. Horváth, “Tuning of dynamic model parameters for adaptive control using particle swarm optimization,” in *Proceedings of the IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems*

ICCC 2022, July 6-9, 2022, Reykjavík, Iceland (A. Szakál, ed.), pp. 197 – 202, IEEE Hungary Section, Budapest, Hungary, 2022.

[28] H. Issa and J. K. Tar, "On the limitations of pso in cooperation with fpi-based adaptive control for nonlinear systems," in *2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES)*, pp. 000181–000186, 2022.

[29] H. Issa and J. K. Tar, "Improvement of an adaptive robot control by particle swarm optimization-based model identification," *Mathematics*, vol. 10, no. 19, 2022.

[30] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[31] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. SIAM books, Philadelphia, 1994.

[32] P. Gahinet, A. Nemirovskii, A. Laub, and M. Chilali, "The lmi control toolbox," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 3, pp. 2038–2041 vol.3, 1994.

[33] G. G. Koch, L. A. Maccari, R. C. Oliveira, and V. F. Montagner, "Robust $\mathscr{H}_\infty$ state feedback controllers based on linear matrix inequalities applied to grid-connected converters," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6021–6031, 2018.

[34] S. Rohani, M. Haeri, and H. Wood, "Modeling and control of a continuous crystallization process Part 2. Model predictive control," *Computers & Chem. Eng.*, vol. 23, p. 279, 1999.

[35] I. Varga, *Model-based parameter estimation and traffic-dependent control of road traffic processes*. PhD Thesis, Technical University of Budapest University of Technology and Economics, 2006.

[36] T. Tettamanti, I. Varga, B. Kulcsár, and J. Bokor, "Model predictive control in urban traffic network management," *In the Proc. of the 16th Mediterranian Conference on Control and Automation, Ajaccio, Corsica, France*, pp. 1538–1543, 2008.

[37] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[38] M. J. Powell, "A method for nonlinear constraints in minimization problems," *Optimization*, pp. 283–298, 1969.

[39] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer, 2011.

[40] A. Grancharova and T. Johansen, *Explicit Nonlinear Model Predictive Control*. Springer, 2012.

[41] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," in *Proceedings of the 27th IEEE Conference on Decision and Control*, pp. 464–465, IEEE, 1988.

[42] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE transactions on automatic control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[43] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles," in *Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301)*, vol. 5, pp. 3741–3746, IEEE, 2002.

[44] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[45] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How, "Distributed robust receding horizon control for multivehicle guidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 627–641, 2007.

[46] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[47] J. Igreja, J. Lemos, and R. Silva, "Adaptive receding horizon control of a distributed collector solar field," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1282–1287, IEEE, 2005.

[48] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari, "Adaptive receding horizon control for constrained mimo systems," *Automatica*, vol. 50, no. 12, pp. 3019–3029, 2014.

[49] A. Lukina, L. Esterle, C. Hirsch, E. Bartocci, J. Yang, A. Tiwari, S. A. Smolka, and R. Grosu, "Ares: adaptive receding-horizon synthesis of optimal plans," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 286–302, Springer, 2017.

[50] C. A. Evangelista, A. Pisano, P. Puleston, and E. Usai, "Receding horizon adaptive second-order sliding mode control for doubly-fed induction generator based wind turbine," *IEEE Transactions on control systems technology*, vol. 25, no. 1, pp. 73–84, 2016.

[51] H. Karabulut, "Physical meaning of Lagrange multipliers," *European Journal of Physics (physics.ed-ph); General Physics (physics.gen-ph)*, vol. 27, pp. 709–718, 2007.

[52] W. Hamilton, "On a general method in dynamics," *Philosophical Transactions of the Royal Society, part II for 1834*, pp. 247–308, 1834.

[53] W. Hamilton, "Second essay on a general method in dynamics," *Philosophical Transactions of the Royal Society, part I for 1835*, pp. 95–144, 1835.

[54] J. Raphson, *Analysis aequationum universalis (Analysis of universal equations).* typis TB prostant venales apud (printed for sale by) A. & I. Churchill, 1702.

[55] T. J. Ypma, “Historical development of the Newton-Raphson method,” *SIAM Review*, vol. 37, no. 4, pp. 531–551, 1995.

[56] C. Kelley, *Solving Nonlinear Equations with Newton’s Method, no 1 in Fundamentals of Algorithms*. SIAM, 2003.

[57] H. Issa and J. Tar, “Speeding up the Reduced Gradient Method for constrained optimization,” *Proc. of the IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI 2021), January 21-23, Herl’any, Slovakia*, pp. 000485–000490, 2021.

[58] H. Redjimi and J. K. Tar, “Approximate model-based state estimation in simplified Receding Horizon Control,” *INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING*, vol. 15, pp. 114–124, 2021.

[59] H. Issa, H. Khan, and J. K. Tar, “Suboptimal adaptive receding horizon control using simplified nonlinear programming,” in *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*, pp. 000221–000228, IEEE, 2021.

[60] J. Acosta, “Furuta’s Pendulum: A conservative nonlinear model for theory and practise,” *Mathematical Problems in Engineering*, vol. 2010, no. Article ID 742894, doi:10.1155/2010/742894, pp. 1–29, 2010.

[61] H. Issa, B. Varga, and J. K. Tar, “A receding horizon-type solution of the inverse kinematic task of redundant robots,” in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 000231–000236, IEEE, 2021.

[62] H. Issa, B. Varga, and J. K. Tar, “Accelerated reduced gradient algorithm for solving the inverse kinematic task of redundant open kinematic chains,” in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 000387–000392, IEEE, 2021.

[63] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–193, International Society for Optics and Photonics, 1997.

[64] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158, IEEE, 2000.

[65] H. M. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas, “A systematization of the unscented Kalman filter theory,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2583–2598, 2015.

[66] Z. Bodó and B. Lantos, “Integrating backstepping control of outdoor quadrotor UAVs,” *Periodica Polytechnica – Electrical Engineering and Computer Science*, vol. 63, no. 2, pp. 122–132, 2019.

[67] H. Redjimi and J. Tar, “The use of multiple components fixed point iteration in the adaptive control of single variable systems,” *In Proc. of the IEEE 17th Intl. Symp. on Intelligent Systems and Informatics (SISY 2019), Subotica, Serbia*, pp. 267–272, 2019.

[68] H. Redjimi and J. K. Tar, “Investigation of noise-sensitivity of a fixed point iteration-based adaptive controller for a pendulum-like electric cart,” in *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 107–112, IEEE, 2019.

[69] H. Redjimi, P. Galambos, J. K. Tar, and K. Széll, “The effects of simultaneous noise and missing information in fixed point iteration-based adaptive control,” in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 1414–1419, 2019.

[70] H. Khan, H. Issa, and J. K. Tar, “Improved simple noise filtering for fixed point iteration-based adaptive controllers,” in *2020 IEEE 20th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 77–82, 2020.

[71] J. Houston, “Economic optimisation using Excel’s Solver: A macroeconomic application and some comments regarding its limitations,” *Computers in Higher Education Economics Review*, vol. 11, no. 1, pp. 2–5, 1997.

[72] E. Naevdal, “Solving continuous-time optimal-control problems with a spreadsheet,” *The Journal of Economic Education*, vol. 34, no. 2, pp. 99–122, 2003.

[73] H. Callen, *Thermodynamics and an Introduction to Thermostatistics (Second Edition)*. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1960.

[74] J. Gram, “Über die Entwickelung reeler Funktionen in Reihen mittelst der Methode der kleinsten Quadrate,” *Journal für die reine und angewandte Mathematik*, vol. 94, pp. 71–73, 1883.

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

[76] P.-S. Laplace, *Théorie Analytique des Probabilités, in: Oeuvres completes de Laplace / publiés sous les auspices de l’Académie des sciences, par MM. les secrétaires perpétuel.* Gauthier-Villars, Paris, 1886.

[77] H. Khan, J. Tar, I. Rudas, and G. Eigner, “Adaptive model predictive control based on fixed point iteration,” *WSEAS Transactions on Systems and Control*, vol. 12, pp. 347–354, 2017.

[78] H. Khan, J. Tar, I. Rudas, and G. Eigner, “Iterative solution in Adaptive Model Predictive Control by using Fixed-Point Transformation method,” *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 12, pp. 7–15, 2018.

[79] H. Issa and J. K. Tar, “Speeding up the reduced gradient method for constrained optimization,” in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 000485–000490, IEEE, 2021.

[80] G. Golub and W. Kahan, “Calculating the singular values and pseudoinverse of a matrix,” *SIAM Journal on Numerical Analysis*, vol. 2, pp. 205–224, 1965.

[81] E. Moore, “On the reciprocal of the general algebraic matrix,” *Bulletin of the American Mathematical Society*, vol. 26, no. 9, pp. 394–395, 1920.

[82] R. Penrose, “A generalized inverse for matrices,” *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 406–413, 1955.

[83] O. Rodrigues, “Des lois géometriques qui regissent les déplacements d’ un systéme solide dans l’ espace, et de la variation des coordonnées provenant de ces déplacement considérées indépendent des causes qui peuvent les produire,” *J. Math. Pures Appl.*, vol. 5, pp. 380–440, 1840.

[84] J. Tar, J. Bitó, L. Nádai, and J. Tenreiro Machado, “Robust Fixed Point Transformations in adaptive control using local basin of attraction,” *Acta Polytechnica Hungarica*, vol. 6, no. 1, pp. 21–37, 2009.

[85] F. Padula and A. Visioli, “Tuning rules for optimal PID and fractional order PID controllers,” *Journal of Process Control*, vol. 21, no. 1, pp. 69–81, 2011.

[86] A. Dumlu and K. Erenturk, “Trajectory tracking control for a 3-DOF parallel manipulator using fractional-order $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ control,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3417–3426, 2014.

[87] L. Bruzzone and P. Fanghella, “Comparison of $\mathrm{PDD}^{1/2}$ and PD^{μ} position controls of a second order linear system,” *In: Proc. of the 33rd IASTED International Conference on Modelling, Identification and Control (MIC 2014)*, pp. 182–188, 2014.

[88] S. Folea, R. De Keyser, I. R. Birs, C. I. Muresan, and C. Ionescu, “Discrete-time implementation and experimental validation of a fractional order PD controller for vibration suppression in airplane wings,” *Acta Polytechnica Hungarica*, vol. 14, no. 1, pp. 191–206, 2017.

[89] E. C. de Oliveira and J. A. Tenreiro Machado, “A review of definitions for fractional derivatives and integral,” *Mathematical Problems in Engineering*, vol. 2014, no. Article ID 238459, p. 6, 2014.

[90] J. Tenreiro Machado and V. Kiryakova, “The chronicles of fractional calculus,” *Fract. Calc. Appl. Anal.*, vol. 20, no. 2, pp. 307–336, 2017.

[91] A. Dineva, J. Tar, A. Várkonyi-Kóczy, and V. Piuri, “Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO systems,” *In Proc. of the IEEE 19th International Conference on Intelligent Engineering Systems, September 3-5, 2015, Bratislava, Slovakia (INES 2015)*, pp. 135–140, 2015.

[92] B. Csanádi, P. Galambos, J. Tar, G. Györök, and A. Serester, “A novel, abstract rotation-based fixed point transformation in adaptive control,” *In the Proc. of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018), October 7-10, 2018, Miyazaki, Japan*, pp. 2577–2582, 2018.

[93] M. Siket, K. Novák, H. Redjimi, J. Tar, L. Kovács, and G. Eigner, “Control of type 1 diabetes mellitus using particle swarm optimization driven receding horizon controller,” *IFAC-PapersOnLine*, vol. 54, no. 15, pp. 293–298, 2021.

[94] H. Khan, A. Szeghegyi, and J. Tar, “Fixed point transformation-based adaptive optimal control using NP,” *In Proc. of the 2017 IEEE 30th Jubilee Neumann Colloquium, November 24-25, 2017, Budapest, Hungary*, pp. 35–40, 2017.

[95] Y. Xu, M. Iwase, and K. Furuta, “Time optimal swing-up control of single pendulum,” *J. Dyn. Sys., Meas., Control*, vol. 123, no. 3, pp. 518–527, 2001.

[96] M. Iwase, K. Astom, K. Furuta, and J. Akesson, “Analysis of safe manual control by using furuta pendulum,” in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 568–572, IEEE, 2006.

[97] B. Cazzolato and Z. Prime, “On the dynamics of the Furuta pendulum,” *Journal of Control Science and Engineering*, vol. 2011, no. Article ID 528341, doi:10.1155/2011/528341, pp. 1–8, 2011.

[98] I. J. Rudas, J. K. Tar, and B. Pátkai, “Compensation of dynamic friction by a fractional order robust controller,” in *2006 IEEE International Conference on Computational Cybernetics*, pp. 1–6, IEEE, 2006.

[99] B. Lantos and Z. Bodó, “High level kinematic and low level nonlinear dynamic control of unmanned ground vehicles,” *Acta Polytechnica Hungarica*, vol. 16, no. 1, pp. 97–117, 2019.

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

[101] J. O. Jang, “Neuro-fuzzy networks saturation compensation of DC motor systems,” *Mechatronics*, vol. 19, no. 4, pp. 529–534, 2009.

[102] Z.-S. Li, X.-Q. Mo, S.-J. Guo, W.-Y. Lan, and C.-Q. Huang, “4-degree-of-freedom anti-windup scheme for plants with actuator saturation,” *Journal of Process Control*, vol. 47, pp. 111–120, 2016.

[103] M. C. Turner and M. Kerr, “A nonlinear modification for improving dynamic anti-windup compensation,” *European Journal of Control*, vol. 41, pp. 44–52, 2018.

[104] 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*, vol. 7, no. 3, pp. 65–80, 1927.

[105] E. Dumetz, J.-Y. Dieulot, P.-J. Barre, F. Colas, and T. Delplace, “Control of an industrial robot using acceleration feedback,” *Journal of Intelligent and Robotic Systems*, vol. 46, no. 2, pp. 111–128, 2006.

[106] B. Czakó, D. Drexler, and L. Kovács, “Continuous time Robust Fixed Point Transformations based control,” *In Proc. of the 2019 IEEE AFRICON Conference, 25-27 Sept. 2019, Accra, Ghana*, pp. 1–6, 2019.

[107] D. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. John Wiley & Sons, New York, 1979.

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

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

[110] H. G. De Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, “UAV attitude estimation using unscented Kalman filter and TRIAD,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4465–4474, 2011.

[111] J. Dunik, M. Simandl, and O. Straka, “Unscented Kalman filter: aspects and adaptive setting of scaling parameter,” *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2411–2416, 2012.

[112] E. Benoît, “Note sur une méthode de résolution des équations normales provenant de l’application de la méthode des moindres carrés à un système d’équations linéaires en nombre inférieur à celui des inconnues (Procédé du Commandant Cholesky)[Note on a Method for Solving Normal Equations Coming from the Application of the Least Squares Method to a System of Linear Equations in Number Less Than That of Unknowns],” *Bulletin Géodésique*, vol. 2, pp. 66–67, 1924.

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

[114] A. Dineva, A. R. Varkonyi-Kóczy, and J. K. Tar, “Combination of RFPT-based adaptive control and classical model identification,” in *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 173–178, IEEE, 2014.

[115] J. Tar, I. Rudas, A. Dineva, and A. Várkonyi-Kóczy, “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*, pp. 35–40, 2014.

[116] J. Tar, J. Bitó, A. Várkonyi-Kóczy, and A. Dineva, *Symbiosys of RFPT-based Adaptivity and the Modified Adaptive Inverse Dynamics Controller, in: Advances in Soft Computing, Intelligent Robotics and Control (Eds. J. Fodor & R. Fullér)*. Springer Heidelberg, London, New York, 2014.

[117] S. Buhmiler, S. Rapajić, S. Medić, N. Duraković, and T. Grbić, “Comparison of derivative-free method and finite-difference method for singular systems,” *Acta Polytechnica Hungarica*, vol. 18, no. 9, pp. 49 – 67, 2021.

[118] L. Cveticanin, G. Mester, and I. Biro, “Parameter influence on the harmonically excited Duffing Oscillator,” *Acta Polytechnica Hungarica*, vol. 11, no. 5, pp. 145 – 160, 2014.

[119] M. Spodniak, L. Főző, R. Andoga, K. Semrád, and K. Beneda, “Methodology for the water injection system design based on numerical models,” *Acta Polytechnica Hungarica*, vol. 18, no. 4, pp. 47 – 62, 2021.

[120] A. Dineva, *Non-conventional Data Representation and Control (PhD Thesis, Supervisors: A.R. Várkonyi-Kóczy & J.K. Tar).* Óbuda University, Budapest, Hungary, 2016.

[121] A. Atinga and J. K. Tar, “Tackling modeling and kinematic inconsistencies by fixed point iteration-based adaptive control,” *Machines*, vol. 11, no. 6, p. 585, 2023.