PhD-FSTC-2019-45
The Faculty of Sciences, Technology and Communication

# DISSERTATION

Defence held on 04/07/2019 in Luxembourg
to obtain the degree of

## DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

## EN *Sciences de l'ingénieur*

by


Sophie KLECKER
Born on 12 July 1990 in Luxembourg (Luxembourg)


# ROBOTIC TRAJECTORY TRACKING: POSITION- AND FORCE-CONTROL

## Dissertation defence committee

Prof. Dr.-Ing. Peter Plapper, dissertation supervisor
*Professor, Université du Luxembourg*

Prof. Dr. Leendert van der Torre, Vice Chairman
*Professor, Université du Luxembourg*

Prof. Dr.-Ing. Holger Voos, Chairman
*Professor, Université du Luxembourg*

Prof. Dr. Gabriel Abba
*Professor, Université de Lorraine*

Prof. Dr.-Ing. Olivier Brüls
*Professor, Université de Liège*

## Abstract

As far as complex dexterous skills are concerned, humans largely outperform machines. As humans naturally fuse information from different sources and adapt their behaviour based on environmental feedback, they achieve a robust and satisfactory performance invariant to dynamic environments or disturbances. Conventionally controlled robotic manipulators are limited to basic monotonous applications in close to ideal circumstances. Their stability and performance are deteriorated in case of adverse conditions like switching constraints or uncertainties. This explains why manual work is predominant in contact-based tasks in manufacturing industry. But tedious, arduous work in non-ergonomic or even hazardous environments, make some tasks unsuitable for human operators. Therefore, the interest in automating these tasks and in expanding the application-areas of robots arose. The goal is to relieve human operators from onerous tasks in conditions which are detrimental to their health and safety. One of the major challenges in this context is the development of adequate control algorithms.

This thesis employs a bottom-up approach to develop robust and adaptive learning algorithms for trajectory tracking: position and torque control. In a first phase, the focus is put on the following of a freeform surface in a discontinuous manner. Next to resulting switching constraints, disturbances and uncertainties, the case of unknown robot models is addressed. In a second phase, once contact has been established between surface and end effector and the freeform path is followed, a desired force is applied. In order to react to changing circumstances, the manipulator needs to show the features of an intelligent agent, i.e. it needs to learn and adapt its behaviour based on a combination of a constant interaction with its environment and preprogramed goals or preferences. The robotic manipulator mimics the human behaviour based on bio-inspired algorithms. In this way it is taken advantage of the know-how and experience of human operators as their knowledge is translated in robot skills. A selection of promising concepts is explored, developed and combined to extend the application areas of robotic manipulators from monotonous, basic tasks in stiff environments to complex constrained processes. Conventional concepts (Sliding Mode Control, PID) are combined with bio-inspired learning (BELBIC, reinforcement based learning) for robust and adaptive control. Independence of robot parameters is guaranteed through approximated robot functions using a Neural Network with online update laws and model-free algorithms. The performance of the concepts is evaluated through simulations and experiments. In complex freeform trajectory tracking applications, excellent absolute mean position errors (<0.3 rad) are achieved. Position and torque control are combined in a parallel concept with minimized absolute mean torque errors (<0.1 Nm).

I would like to thank all the people whom I met or who accompanied me during my PhD-time.

Thank you for always supporting me, in one way or another.

Without you these past 4 years would not have been the same.

Villmools Merci!

## List of Figures

## List of Tables

# Table of Contents

# 1 Introduction

The purpose of this first chapter of the thesis is to put the work into its context.
The first part deals with the framework of this project.
The second part details the objective of the thesis.
The outline of the thesis-report is the subject of the final part.

## 1.1 Context

This part presents the context of the presented work in four steps: The first paragraph briefly establishes the motivation for automation of complex manufacturing tasks. The second paragraph justifies the interest in the research work of the thesis from an industrial as well as from a scientific point of view. The third paragraph presents the vision and the mission of the research on a global level and on the thesis-scale, respectively. The final paragraph presents in general terms the methods applied in this research and paves the way for the following chapters.

### 1.1.1 Motivation

Human skills and performance are unrivalled in complex dexterous manipulation. Humans naturally fuse information from different sources and adapt their behaviour based on environmental feedback. As far as adaptive handling is concerned, humans outperform robotic manipulators and machines largely. Applied to manufacturing industry, this explains why human workers are predominant in complex manufacturing operations in factories (Faber, 2015) (Perzylo, 2019). Industrial state-of-the-art comprises manual work for most contact-based manufacturing tasks including surface finishing processes. By their nature, human operators achieve a robust, satisfactory performance invariant to dynamically changing constraints and uncertainties. However, tedious and arduous work in non-ergonomic or even hazardous environments, make some tasks unsuitable for human operators. Therefore, the interest in automating these tasks arose. One of the major challenges in automating these tasks is the development of adequate control algorithms for a robust performance of the robotic manipulators especially for the case of dynamically changing constraints. Ideally, the robotic manipulators take advantage of the experience and knowledge of human operators.

### 1.1.2 Interest

The automation of complex manufacturing tasks in general and surface finishing processes in particular is of significant interest to both the industrial and the scientific communities.

Automation or partial automation of surface finishing processes of complex freeform workpieces is of high interest to the industrial community. Both, workpieces and processes become more and more complex (Klocke, 2014). Despite the introduction of robotic manipulators in factories, their areas of application are limited to monotonous processes in stiff environments. Only in basic tasks in close to ideal circumstances, satisfactory performance, in the sense of acceptable deviations from the desired values qua position and force, is achieved (Qin, 2016). Manual processes are current industrial state-of-the-art for complex tasks in the presence of varying constraints and uncertainties. Due to their time- and cost-intensive nature, surface finishing processes are the bottleneck of the considered industry. Different studies suggest shares of up to 30-50% of the entire manufacturing time and up to 40% of the total cost (Dieste, 2013) (Feng, 2000) (Lee, 2001) (Lin, 2014) (Pagilla, 2001) (Robertsson, 2006) (Roswell, 2006) (Wilbert, 2012) (Yixu, 2012).

From a scientific point of view, multiple challenges are related to the considered automation problem. First, the transfer of knowledge from human workers to robots has to be assured, i.e. the human skills have to be translated in robot skills in a way that takes optimally advantage of the human expertise. A teaming of human and robot skills and intelligence is desirable (Johnson, 2019). Second, the process has to be controlled. This requires a control concept for the robotic manipulator. Control algorithms have to be designed to guarantee a stable and robust behaviour of the robotic manipulator. The control problem is a trajectory tracking problem which aims in a first phase, to follow a complex freeform surface in a discontinuous manner. The resulting switching constraints impede the stability of the contact-based task (Liberzon, 2003). Further factors affecting the robust behaviour of the system are internal and external uncertainties. The latter are environmental disturbances whereas the former include uncertain or unknown robot models (Sünderhauf, 2018). These make an adaptive algorithm necessary. A controller independent from a priori known robot models 1) ensures stability with respect to unknown robot parameters and 2) allows implementation on different robot types. Through a combination of constant interaction with its environment and preprogrammed elements, the robotic manipulator needs to learn and adapt its behaviour. In this context, the interest from the artificial intelligence research community can be explained. In a second phase, once contact has been established between workpiece and end effector, the freeform path is followed, a desired force is to be applied. The desired performance is defined as minimized mean position and force errors. The reduction of average deviations between desired and current position and force signals over the total process time guarantees that 1) the desired trajectory is followed as closely as possible with minimal lifting of or digging into the surface and 2) the applied forces are adapted to the process.

### 1.1.3 Vision and Mission

The final vision motivating this research is an automated surface finishing process or similarly complex manufacturing process applicable in an industrial milieu. The goal is to create value through relieving human operators of onerous tasks in conditions which are detrimental to their health and safety while still taking advantage of their know-how (Johnson, 2019) (Kaihara, 2018). More specifically, the envisioned idea englobes two phases. In a first phase, the human operator shows the desired behaviour to the robot arm through kinaesthetic teaching. In a second phase, the robotic manipulator uses the recorded behaviour to achieve the desired surface finishing results while flexibly adapting to environmental changes. Through a combination of robust and bio-inspired control algorithms, stability and performance are ensured even in adverse conditions.

This thesis puts the focus on the development of the position and force control algorithms (Barto, 2019) (Sünderhauf, 2018). They combine robustness and adaptability with independence from a priori known robot model parameters for reduced absolute mean position and torque errors.

### 1.1.4 Bottom-up Approach

The global problem addressed in the thesis is the automation of constrained manufacturing tasks. Humans being highly proficient in these tasks requiring adaptive position- and force-control, the aim is to find inspiration in the human workers' behaviour and to translate their capabilities into robot skills. Therefore in a first step the human approach to perform a constrained manufacturing task is analysed. Based on the observations, an abstraction of the comprehensive problem as well as its subdivision into two bottom-up sub-problems is deduced. The abstracted challenge is defined as the simultaneous position- and force-control of an industrial robot arm in the presence of uncertainties, switching constraints and friction. The subdivision of the global problem into 'Path Following', i.e. position control and 'Path Following + Application of a Force', i.e. combined position and force control is illustrated in Figure 1. Breaking down the global problem into sub-problems allows to better understand and address the individual challenges as well as their interactions. As a visualization for a surface finishing process, this subdivision means that, first, the surface is followed and then the force is added to perform the manufacturing. The work of this thesis consists in the development of adequate algorithms for addressing the defined control problem. The chosen approach is bio-inspired which also translates in bio-inspired algorithms.

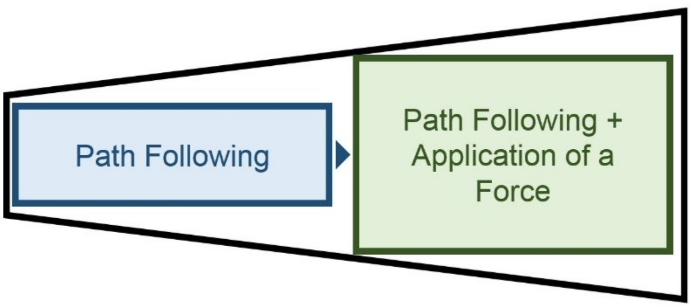



Figure 1: Subdivision of the comprehensive problem for automating constrained tasks into bottom-up sub-challenges 'Path Following' and 'Path Following + Application of a Force' .

## 1.2 Objective

Automated systems and robots have proliferated in basic monotonous industrial processes due to their repeatability and accuracy. As a consequence of their unequalled performance in dexterous and adaptive manipulation, human operators are still prevailing in surface finishing tasks. These processes include tedious work in hazardous environments which are not suited for humans. Therefore the need to replace humans by robotic manipulators for these tasks emerged. The respective technological challenges can be summarized as developing control algorithms for discontinuous trajectory tracking of freeform geometries (Giusti, 2018). The presence of switching constraints, uncertainties and potentially unknown robot model parameters impacts the stability as well as the performance of the position and force controllers.

Conventional controllers present the advantages of being field-tested and simple to implement. More advanced control algorithms focus on increasing stability in the presence of uncertainties and disturbances. Despite high robustness and accuracy, flexibility and performance compared to human operators are poor (Adar, 2016) (Cervantes, 2001) (Karayiannidis, 2012). The aim of human mimicking concepts on the other hand is to copy one-to-one the approach of human operators. Despite a human-like behaviour of the automated system, robustness and usability are limited (Barto, 2019).

A control algorithm for combined position and force control for complex trajectory tracking applications has to guarantee robust and accurate performance. In order to react to changing circumstances, it also needs to show the features of an intelligent agent, i.e. it needs to learn and adapt its behaviour based on a combination of a constant interaction with its environment and preprogramed goals or preferences. Consequently, robust control algorithms are combined with bio-inspired features. The aim is to take advantage of the expertise of the human operator (Johnson, 2019). Through kinaesthetic teaching, the human operator demonstrates and transfers his process know-how to the robot. Further inputs to the system include real-time feedback from position- and torque-sensors as well as process-related preferences and goals.

The objectives of the developed algorithm are

1) Process performance defined as minimization of position and torque errors. More precisely, the deviations between desired and measured position and torque signals averaged over the total process time are kept below a threshold (<0.3 rad and <0.1 Nm).

2) Robustness with respect to unknown robot parameters defined as independence of robot models (Sünderhauf, 2018).

Finally, the research will result in an automated complex manufacturing process applicable in a real world-environment.

## 1.3 Outline

This part presents the structure and contents of the rest of the report. Figure 2 illustrates the connections and relations of the seven chapters.

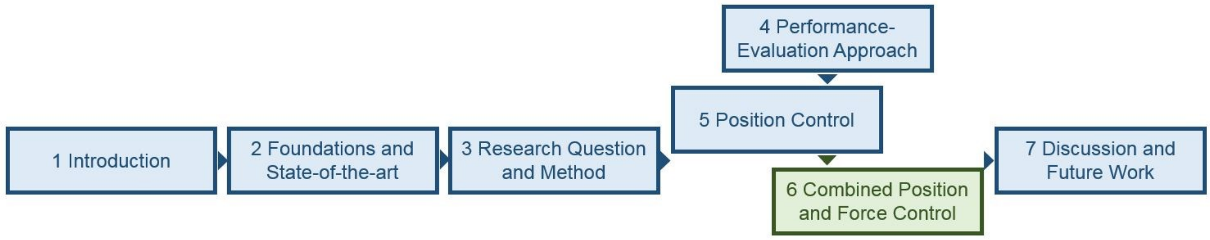



Figure 2: Structure of the thesis.

The introductory chapter 1 establishes the context of the presented work, it describes the addressed problem. The framework, vision and objective of the research are defined and elaborated on.

The second chapter establishes what is already known about the addressed problem and points out shortcomings that require further research. State-of-the-art publications in related domains are categorized according to their contributions. From the literature survey, the scientific gap which is going to be addressed in the thesis is deduced and the work presented as a continuation of the cited previous research efforts.

Chapter 3 explains how the gap deduced in chapter 2 will be filled. The research question, hypothesis and intended methodology are described.

In chapter 4, the setups for the numerical simulations and the experiments are detailed. These setups will be used for validation purposes in chapters 5 and 6.

The fifth chapter is dedicated to the first sub-challenge, i.e. path following or position control. The suggested control algorithms are developed, their stability analysed and their performance validated using the strategies introduced in chapter 4.

Chapter 6 is dedicated to the extension of chapter 5, namely to the second sub-challenge, i.e. path following + application of a force or combined position and force control. The suggested control algorithms are developed, their stability analysed and their performance validated using the strategies introduced in chapter 4.

In the final chapter, the presented research and its contributions are discussed. Also, other possible avenues for this thesis are considered. Naturally, this process allows the discussion of potential areas for further work.

The thesis ends with a summary.

# 2 Foundations and State-of-the-art

In previous years and decades, research has already been performed on different aspects of the topic addressed in this thesis. The aim is to build on and take advantage of the progress and experiences of previous research works. During the analysis of the state-of-the-art, however, three types of shortcomings were revealed. First, potential for optimization is found in the developed algorithms for position control, estimation of unknown parameters and combined position and force control. Second, works combining multiple promising concepts are scarce. Third, a lack of publications describing industry-relevant applications is noticed. From the description of these shortcomings, the scientific questions at the basis of this work are deduced in order to contribute in the most optimal way to the global research. In this sense, the suggested research can be seen as a continuation of previous works.

The purposes of this second chapter of the thesis are first to establish what is already known on the topic, second to deduce potential gaps where need for further research persists and third to establish the presented research as a continuation of previous scientific work.

The first part, 2.1, clarifies concepts related to the presented research which are already well established in and by the scientific community. Terms which will be used throughout the thesis are explained and conventional controllers which will be used for comparison purposes in later chapters are described.

The subsequent parts, 2.2, 2.3 and 2.4 present a literature survey. Related relevant publications on position control, approximation of unknown robot parameters and combined position- force-control are cited.

In 2.5, the literature survey is summarized and the areas for further research addressed in this research work are deduced. In this part, it is detailed how this thesis fits in the cited literature.

## 2.1 Foundations

This part is dedicated to clarifying concepts related to the presented research which are already well established in and by the scientific community. Before related scientific works are cited and the introduced research is presented as their continuation, the most relevant known concepts are announced and described in detail. This is done in two steps: In 2.1.1 terms and ideas used and referred to throughout the thesis are formulated. In 2.1.2 two illustrative examples of basic control algorithms are spelled out. These conventional controllers will be used for comparison purposes in chapters 5 and 6.

### 2.1.1 Definition of Key Concepts

In this paragraph, concepts which are important for and used throughout the thesis are introduced in a thematically clustered fashion and clarified.

For automation purposes, robots are predominantly used to support human operators through replicating their behaviour. The field of robotics deals with the operation and control of robots.

Robotic manipulator

A robotic manipulator also known as a serial robot, is an arm-like programmable mechanical device with a number of degrees of freedom that is i.a. used for manipulation purposes. The mechanism is made up of links interconnected by joints. The kinematic chain is terminated by a task-specific end effector.

Robotic manipulators are illustrative examples of nonlinear and time varying MIMO-plants (multiple inputs multiple outputs). They are subject to internal and external uncertainties. System-inherent uncertainties include potentially unknown dynamic and kinematic model parameters. The system's implementation in and interaction with the environment causes external disturbances which may impact its stable performance.

Application areas are highly diverse and their discussion is out of the scope of this thesis. They range inter alia from manufacturing processes over assembly tasks to robot-assisted surgery (Siciliano, 2008) (Spong, 2006).

Constrained movement

The interactions between end-effector and surrounding environment engender the former's movement to be constrained by the latter i.e. the robot arm cannot move freely in all directions (Jankowski, 1999). Constraints include natural constraints due to the specificities of the environment as well as artificial constraints due to and characteristic of the desired task.

Switching constraints

A change in constraints between the end-effector of the robot and the environment brought about by the successive discrete or continuous phases of e.g. assembly or manufacturing processes, results in a switched nonlinear system, undermining the stable behaviour (Liberzon, 2003). Common examples are transitions from free space- to in-contact-situations and vice-versa.

Programming by Demonstration

Programming by Demonstration, known under the abbreviation PbD, teaches a robot by showing the desired behaviour rather than by writing commands in a programming language. It is a form of imitation learning which seeks to acquire skills from observing demonstrations (Argall, 2009) (Calinon, 2009). The use of demonstrations as information-form contributes to the unambiguous communication of a task. They do not require knowledge of a language and they are applicable also for tasks which are difficult to express in other forms like speech or programming language (Duan, 2017).

Kinaesthetic teaching

Kinaesthetic teaching is a type of PbD (Rozo, 2016). Here, the human instructor physically interacts with the robot as he grabs and displaces the robot's body parts and operates its tool in space and time (Giusti, 2018). This is enabled by the robot compensating for the effect of gravity on its limbs as the human holds the robot arm and moves it through the task. This procedure allows the user to move the robot arm without feeling its weight and without feeling the motors in the articulations, while the sensors are used to record the movement (Racca, 2016).

Trajectory tracking

Trajectory tracking aims at following a predefined desired joint path, i.e. the robot end-effector is expected to follow a particular path within the workspace of the robotic manipulator.

The aim of control engineering is the design of dynamical systems with desired behaviours. The development of a control model uses a control action to ensure the performance of the considered continuously operating process or system matches as closely as possible the targeted performance. A controller is developed to induce corrective behaviour while guaranteeing stability, minimizing delay and overshoot. Overshoot is defined as a signal surpassing the desired value. Ideally, the controlled system would not present any instability, delay or overshoot (Slotine, 1991) (Unbehauen, 2008).

Position control

Position control is described as the control of the following of predefined positions by a robot end-effector. Positioning is a synonym to goal-reaching, namely the getting to a target position. Trajectory tracking or path following is a continuous form of positioning, that is to say the uninterrupted following of successive desired positions. In both cases of position control, the torques resulting in the desired positions have to be computed. These torques are the control output of the position controller (Craig, 2005) (Siciliano, 2008) (Toibero, 2011).

Force control

Force control addresses contact situations between the robot end-effector and its environment, i.e. the control of predefined interaction forces. Focussing on industrial applications, the most relevant use cases are grinding, assembly and friction stir welding tasks. Direct force control uses a direct explicit force/torque feedback control loop (Siciliano, 2008).

Feedback control

Feedback control is also known as closed loop control. Desired variables are defined a priori based on a model of the process. Current variables are commonly measured with sensors and detectors. Through a feedback loop the current process variable is fed back to the controller whose role is the minimization of the discrepancy between current and desired process variable. The related controller output depends on feedback from the process (Di Steffano, 1967) (Mayr, 1970).

Adaptive control

A self-adapting control law is defined as adaptive controller. In contrast to robust controllers which do not adapt themselves and guarantee stability for parameters fluctuating only within a priori defined bounds, adaptive controllers readjust themselves and can accommodate varying or uncertain parameters (Åström, 2008).

Model-free control

Model-free control does not require a model of the controlled system, i.e. neither the robot model nor its approximation are used for constructing the control input. The main advantages are the simple design and implementation, fast tuning and robustness with respect to parameter uncertainties and modelling inaccuracies.

Besides basic on-off-controllers, PID-controllers are examples of conventional model-free controllers. Due to their ease of design, simple structure and robustness, conventional linear fixed-gain, model-free PID-controlled robotic manipulators have been predominant in industrial settings since their introduction in 1940 (Adar, 2016) (Unbehauen, 2008). According to a survey conducted in 2002 (Desborough, 2002), PID-controllers made up for more than 97% of regulatory systems in process industry. Using the passivity property, asymptotic stability has been proven in local (Arimoto, 1996), semiglobal (Ortega, 1995) as well as in global sense (Arimoto, 1994) (Arimoto, 1994b).

Model-based control

In contrast to model-free controllers, model-based algorithms make use of the robot model or its estimation for computing the controller output. In an effort to accommodate the particular specificities of the considered robotic manipulators, the interest in these model-based controllers emerged. They advanced to the predominant approach for control problems in robotics (Sünderhauf, 2018). A prime example of conventional model-based control algorithms is the computed torque method that was developed in the 1970's. (Markiewicz, 1973) analysed the computed torque drive method as a control concept able to deal with the nonlinearities inherent to the manipulator and compared it to a conventional position servo drive method. (Qu, 1991) investigated the robustness of robot control by the computed torque law. (Spong, 2006) lists relevant publications in the field. The main motivations for using model-based control are its high expected performance tailored to the system at stake.

Artificial intelligence also known under the abbreviation AI focuses on research on computational agents that act intelligently.

Agent

An agent is defined as an actor perceiving, reasoning and acting in an environment. Its outputs, i.e. actions depend on inputs, i.e. on the agent's aptitudes, on its knowledge about the environment, itself and its goals or preferences as well as on the records of experiences of previous interactions with the environment. How the outputs are derived from the inputs is an intensive research topic.

In this work, a robotic manipulator as a pairing of actuators, sensors and a computation-unit in a physical environment is defined as agent (Poole, 2017).

Intelligence

An agent is classified as intelligent when first, its actions are aligned with its goals or preferences, second, it adapts to changes in its goals or in its surroundings and third, it learns from experience (Poole, 2017).

Since its introduction in the 1950s, artificial intelligence, commonly known under the abbreviation AI, has been researched on. However, interest, developments and applications exploded decades afterwards with the advancements in informatics components and computational power. Artificial intelligence which focuses on a singular task, for which it has been programmed for, is denominated weak or narrow AI. This form of intelligence is distinct from a general intelligence as we know it from humans (Blanchot, 2019).

Learning

Learning is defined as an agent's capacity to improve its future by adapting its behaviour based on past experiences. Learning in artificial systems as opposed to humans or natural biological systems in general is also referred to as machine learning.

Learning is categorized as supervised when the classification, i.e. what has to be learned is provided to the agent. In unsupervised learning on the contrary, the agent has to find the classification itself without external guidance.

Learning is further categorized as offline when all training samples are provided to the agent a priori, i.e. before acting. In online learning on the contrary, training samples are provided while acting (Poole, 2017) (Blanchot, 2019).

Reinforcement learning

Reinforcement learning is based on one of the first concepts of AI introduced by Alan Turing in 1948 (Barto, 2019). The agent learns through interactions with its environment while trying to maximize a specified cumulative reward (Bertsekas, 1996) (Kaelbling, 1996) (Sutton, 1998). Learning through interaction is the essence of this experience-driven autonomous learning and reward-driven behaviour. The agent autonomously interacts with its environment via actions which are consecutively evaluated by a reward function and awarded either a reward, positive reinforcement signal or a punishment, negative reinforcement signal. Through observing these consequences of its actions, the agent learns to adapt and optimize its behaviour improving over time, e.g. through trial and error. Every transition to a new state of the environment, provides feedback in the form of reinforcement signals to the agent thereby yielding information which the latter uses to update its knowledge. In this sense, the agent interacts with and learns from its surroundings. The optimal action-sequence being defined by the feedback provided by the environment and the scalar rewards, reinforcement learning shows similarities with optimal control (Arulkumaran, 2017). Reinforcement learning shows characteristics of supervised as well as of unsupervised learning (Barto, 2019). Reinforcement signals appear after a series of actions and lead to a credit assignment problem of learning which actions are accountable for which reinforcement signals (Poole, 2017).

Traditionally, agent interactions in reinforcement learning are characterized by:

- a state signal describing the state of the agent's environment,
- an action signal through which the agent influences its environment and
- a reward signal providing feedback of the outcome of the taken action.

Figure 3 illustrates these interactions in the form of a state perception-action-feedback learning loop.

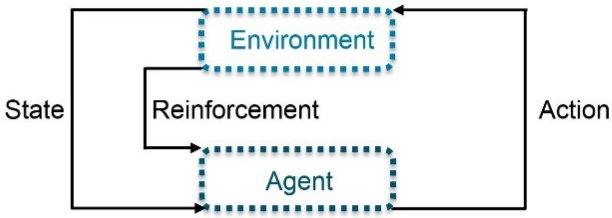



Figure 3: The principle of Reinforcement Learning.

The main advantages of the concept are the online adaptability and the fact that full knowledge of the system and its environment is not required. Reinforcement learning has been proposed as model-free as well as as model-based algorithm. In a principled mathematical framework, it can be described as a Markov decision process (Arulkumaran, 2017). Its analogy with the functioning of the human brain and neuroscientific theories qualifies reinforcement learning as bio-inspired concept (Barto, 2019). The basic concept of reinforcement learning combining learning from experience with rewards is inspired by behaviourist psychology, i.e. the aspiration of living beings for survival and growth (Lewis, 2009). The goal for the agent is to learn optimal behaviour over time. Reinforcement learning aims to bridge the gap between bio-inspired learning techniques, adaptive and optimal control (Khan, 2012). For solving reinforcement learning problems, value function, policy search and hybrid actor-critic methods have been suggested (Arulkumaran, 2017). Reinforcement learning was successfully applied i.a. to playing GO (Silver, 2017) and robot control (Peters, 2008) (Hester, 2011).

Bio-inspired design

Bio-inspired engineering aims for human-made solutions inspired by biological solutions, taking advantage of the strength of natural evolution. Recently, bio-inspired engineering has celebrated resounding successes due to its innovative, relatively simple and intuitive solutions to highly complex problems. Bio-inspiration is applied to a wide range of areas related to robotics. The application of bio-inspired or intelligent algorithms is twofold. First, the structure of algorithms can be inspired on that of biological, natural systems. Second, the planned behaviour can be bio-inspired, i.e. imitating the behaviour of natural systems and thereby profiting from their knowledge and expertise (Passino, 2005). Nature-inspired algorithms are i.a. popular in clustering. Clustering algorithms based on the principles observed in brain circuits: artificial neural networks (Figueira, 2011), in the migration, extinction and emergence of biological species: biogeography based (Hamdad, 2013) as well as in fish schools (Huang, 2013), bird flocks (Barve, 2013) (Cui, 2006) (Cui, 2006b) (Forestiero, 2013) and ant colonies (Elkamel, 2015) (Vaijavanthi, 2011): swarm intelligence was subject of research efforts.

Neural Networks

In mathematical modelling, neural networks are artificial bio-inspired networks mimicking the natural mammalian central nervous system. Their composition of a significant number of simple but highly interconnected units, i.e. collaborating neurons loosely mimics the structure and functioning of the human brain. Since their introduction in the 1940s (McCulloch, 1943), these information processing paradigms are widely used among others for the estimation of functions dependent on unknown parameters and for control purposes. They are a prominent choice not least due to their parallel architecture, their impressive learning capacity despite low computational costs, their fault tolerance and possibility to be implemented in real-time applications (Liu, 2013).

Billions of neurons can be found in the human brain which output and transmit signals to neighbouring neurons or receptors, a process known as synapse. Preceding the synapse, an electrical pulse known as spike travels through the neuron's axon during the depolarization phase. This spike is triggered when the sum of received inputs reaches a threshold value. The synaptic adaptability and plasticity is primordial in the human's capability to memorize and learn from experience (Passino, 2005).

This natural synaptic learning is translated in adaptive artificial neural networks by implementing an adaptive rule, i.e. by allowing the weights $w$ to change over time ($\dot{w} \neq 0$). A simplified, schematic version of a natural neuron is depicted in Figure 4 (left) while Figure 4 (right) shows the structure of an artificial neuron. The output of the latter can be expressed as:

$$y_i = \sum_{j=1}^{n} w_{ij} h(x_j) \tag{1}$$

where $i$ is the number of the considered neuron, $h$ is the activation function, $n$ is the number of inputs, $w_{ij}$ is the weight from unit $i$ to unit $j$ and $x_j$ is the input into unit $j$.

Radial Basis Function Neural Network

Radial Basis Functions were presented as technique for interpolation in multidimensional space by (Micchelli, 1986). Radial Basis Functions, known under the abbreviation RBF are real-valued functions whose output-value is solely dependent on the distance from a centre $c$: $h(x,c) = h(\|x - c\|)$ where the distance function, the norm typically is the Euclidean distance. The RBF used here is a Gaussian type.

Their implementation as activation functions in neural networks is known as a 3-layer network under the acronym RBF-NN. For each node the output is defined based on its input or set of inputs. These radial basis neural networks, introduced in 1988 (Broomhead, 1988), have the ability to learn arbitrary mapping and count systems control, classification problems and function estimation among their primary application areas.

RBF-NNs are feedforward neural networks which means they allow signals to travel one way only, in this case from input to output. A typical RBF-NN has three layers: an input layer, a hidden layer and an output layer. Its first layer, i.e. the input layer performs a nonlinear transformation mapping the input signals to the hidden layer. The single hidden layer consists of an array of computing units, i.e. hidden nodes which are activated by RBF activation functions. The output layer consists of linear summing nodes which allow the network's output to range over a significant range of values.

Based on the inputs, i.e. the raw information that is fed into the neural network, a radial basis function, usually of Gaussian type activates the neurons of the hidden layer and produces the output. The output layer is a linear weighted combination of RBFs of the inputs and internal parameters. Characteristics of RBF-NNs with Gaussian activation functions are that their output is maximal at input values close to center $c$, tailing off with increasing distance between input and center. Further, the sharpness of the network is expressed as the standard deviation of the Gaussian function at stake. Training takes place via the choice of hidden units number as well as of center and sharpness parameters.

Compared to multilayer neural networks, their structure is less coherent with the natural neural network but their advantages include: faster convergence, more straightforward training and analysis due to a simpler topology and easier implementation due to fewer interconnections (Bass, 1994) (Van Cuong, 2016). (Park, 1991) (Tao, 2016) (Yu, 2014) i.a. have shown the universal approximation-capabilities of RBF-NNs. Additional advantages of RBF-NNs are the relaxation of the need for a priori knowledge about unknown parameters as well as their ability to accommodate a wide range of uncertainties.

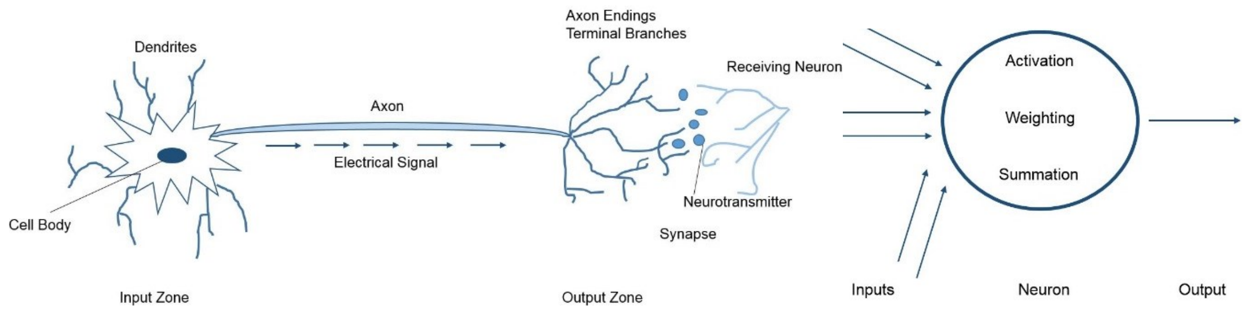



Figure 4: Schematic representation of a natural neuron (left); Structure of an artificial neuron (right).

### 2.1.2 Conventional Controllers

Two illustrative examples of conventional controllers are described below. A PID-controller is chosen as an illustrative representative of conventional model-free controllers while a Computed Joint Torque-controller represents model-based schemes.

<u>PID-Controller</u>

This model-free controller combines a proportional, an integral and a derivative controller-part as shown in Figure 5. The PID-controller is a common linear controller whose transfer behaviour is based on the parallel arrangement of ideal P-, I- and D-elements. The control output of the error-driven PID-controller is:

$$\boldsymbol{u} = -\boldsymbol{D_p}\boldsymbol{error} - \boldsymbol{D_i}\int_0^t \boldsymbol{error}\,dt - \boldsymbol{D_d}\dot{\boldsymbol{error}} \tag{2}$$

where $\boldsymbol{D_p} = d_p\boldsymbol{I_n}, \boldsymbol{D_i} = d_i\boldsymbol{I_n}, \boldsymbol{D_d} = d_d\boldsymbol{I_n}$ with $d_p, d_i, d_d$ constants, $\boldsymbol{I_n}$ an nxn identity matrix and $\boldsymbol{error}$ the difference between actual and desired control variable.

$$\boldsymbol{error} = \boldsymbol{actual} - \boldsymbol{desired} \tag{3}$$

Adequate controller gains $\boldsymbol{D_p}, \boldsymbol{D_i}, \boldsymbol{D_d}$ are essential for a satisfactory performance of the system. They are chosen a) through trial and error, manual tuning, b) using empirical approaches like Ziegler-Nichols method or c) based on process models. Setting to zero one or two of the gains, leads to special cases of the PID-controller: P-, PI- or PD-controllers.

The advantages of the PID-controller include ease of design, simple structure and robustness. Its disadvantages include lack of process knowledge, limited adaptability due to fixed gains as well as poor performance for nonlinear systems (Slotine, 1991) (Spong, 1992) (Unbehauen, 2008).

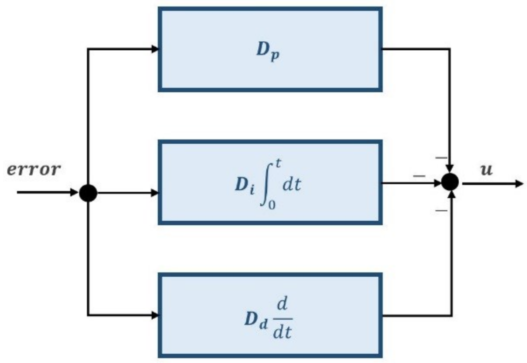



Figure 5: PID-controller.

<u>Computed Joint Torque-Controller</u>

Representative for conventional model-based controllers, a variant of computed joint torque-controller with the control as in eq. (4) is used. Figure 6 illustrates this model-based controller.

$$\boldsymbol{u} = \boldsymbol{M}(\boldsymbol{actual})(\ddot{\boldsymbol{desired}} + \boldsymbol{D_p}\boldsymbol{error} + \boldsymbol{D_d}\dot{\boldsymbol{error}}) \tag{4}$$

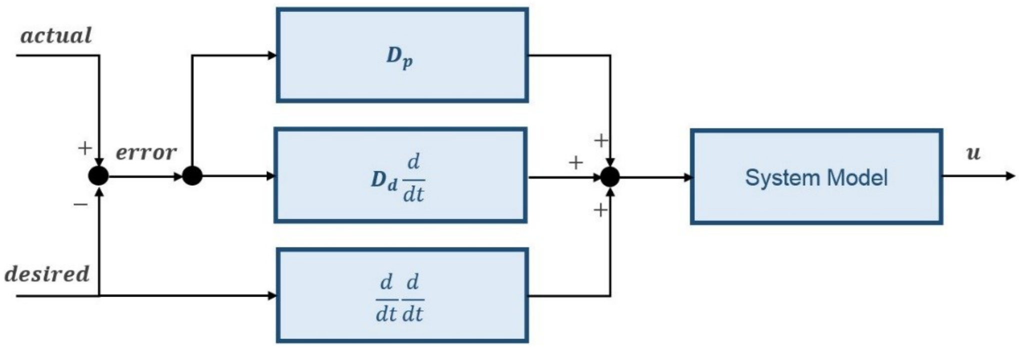



Figure 6: Computed Joint Torque-controller.

Computed-torque control, also known as inverse dynamic control, is an effective model-based strategy, for example in trajectory tracking of robotic manipulators. The required control torque is computed as a function of desired and actual control variables while taking into account the system dynamic model. In contrast to PID-controllers, computed torque-controllers can handle nonlinearities as nonlinear system dynamics are compensated through feedback linearization. For the ideal case of known system models, linear, decoupled error dynamics with asymptotic stability are achieved. In practice, however, a trade-off between minimal response-time and overshoot has to be made. A further effect of physical actuator-limitations is a risk for instabilities. In addition, uncertain or changing model parameters negatively impact the performance and stability of the controller (Chen, 2012) (Craig, 2005) (Luh, 1980) (Markiewicz, 1973) (Peng, 2009) (Siciliano, 2008) (Siciliano, 2010) (Wang, 2011).

## 2. 2 State-of-the-art: Position Control

Together with pick-and-place tasks, trajectory tracking is the most common type of automation tasks (Giusti, 2018). Position control, i.e. the following of pre-defined positions by a robot end-effector has therefore triggered the interest of the scientific community. The interactions between manipulator and surrounding environment as required by these trajectory tracking applications engender the former's movement to be constrained by the latter (Jankowski, 1999). A change in constraints between the end-effector of the robot and the environment brought about by the successive process-phases results in a switched nonlinear system, potentially undermining the stable behaviour (Liberzon, 2003). The system's implementation in and interaction with the environment causes external disturbances which impact the stable system performance. In addition to external disturbances, system-inherent uncertainties deteriorate the overall system's performance. They include ambiguous system parameters. The combination of these aspects impacting the stable controller-performance explains the scientific community's continuing interest in trajectory tracking control problems.

As far as model-free controllers are concerned, trajectory tracking problems were addressed by PID-controllers (Kawamura, 1988) (Qu, 1991) (Wen, 1990). Satisfactory performance in terms of position error can be achieved for basic manipulators and applications. The performance however is highly dependent on the gains and their tuning (Karayiannidis, 2012). The main drawback of this type of purely error-driven control is suboptimal performance. Their constant fixed parameters as well as their linearity make it hard to cope with either nonlinear, time-varying systems or disturbances. The lack of flexible adaptability and the impossibility to increase gains arbitrarily due to actuator limitations as well as the occurrence of instabilities and noise sensitivity (Kuc, 2000) (Siciliano, 2008) limit the application areas. (Alontseva, 2019) investigated surface tracking control for robot manipulators. The authors used linear models of robot tool dynamics. Finally, researchers and practitioners in the field agree that conventional controllers are not suited for freeform trajectory tracking in general and processes with highly nonlinear, coupled robotic systems in particular (Adar, 2016) (Cervantes, 2001) (Kuc, 2000) (Lee, 2001) (Siciliano, 2008).

As far as model-based controllers are concerned, sliding mode control, SMC, is a prominent robust control strategy (Ginoya, 2015) (Liu, 2011). Being invariant to both external and internal disturbances, the variable structure control method is chosen for controlling systems with strong nonlinearities as well as parametric and modelling uncertainties. A high-speed switching control law is used to first, guide the system's state trajectory onto a user-defined sliding surface and to second, keep it on this sliding surface for subsequent times. This behaviour is depicted in Figure 7. (Yao, 1998) applied a sliding mode controller to robot manipulators. (Hackl, 2009) used sliding mode control for position- and velocity-tracking of robot joints. (Jasim, 2013) suggested to control a switched nonlinear system, i.e. a robotic manipulator with continuously switching constraints with an adaptive sliding mode controller. Although SMC is invariant to internal and external disturbances, two major drawbacks limit the performance or application areas for this type of variable structure control. The first drawback is chattering. These undesired oscillations result from imperfect control switchings. Keeping the system's state trajectory on the smooth sliding surface would require infinitely fast switching. As in reality, switching is restricted to a finite frequency, an oscillation around the switching surface is observed. The second drawback is the need for a priori knowledge of function bounds.

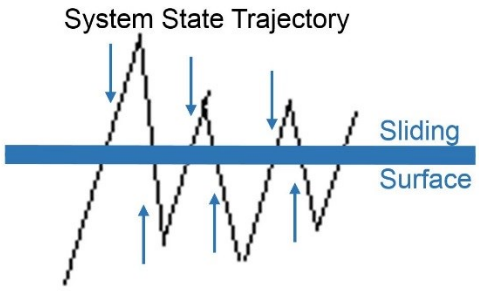



Figure 7: The principle of SMC - Sliding Mode Control.

To improve the performance of SMC or conventional control algorithms in general and to focus on so far non-addressable control problems, combinations with robust, adaptive or intelligent controller-extensions were suggested. A combination of SMC and adaptive elements was applied to trajectory tracking for freeform grinding by (Klecker, 2016) (Klecker, 2016b). (Qu, 1995) suggested the combination of a conventional control law, either PD or computed torque law, with a robust estimator for a trajectory tracking application. Fuzzy logic-based extensions were suggested to overcome the drawbacks of SMC (Li, 2010) (Li, 2015) (Lian, 2013) (Roopaei, 2009).

A concept which triggered the interest of the dedicated scientific community is bio-inspiration. Bio-inspiration has been used in domains like control and then primarily for feedback control as well as in trajectory tracking. The fact that humans effortlessly control the poses of and the forces applied by their hands and tools, explains the interest in mimicking the human behaviour in the field (Kronander, 2014). An analogy with a human operator had also been established by (Navarro-Gonzalez, 2015). Key requirements pursued in intelligent control are often inspired on human capacities and comprise intuitive, robust, adaptive behaviour and fast training. (Tramper, 2013) studied the human's combination of anticipation and ongoing feedback control for contour following tasks. The importance of feedback was also extensively stressed by (Tabot, 2014). (Huang, 2003) investigated how a contact force is adapted when it suddenly deviates from the expected or desired force. In (Yin, 2004) the hierarchical architecture of the human tactility was imitated to make a robot track an unknown surface.

In this context, the suggestion of bio-inspired controller-extensions is a logical consequence. Their key advantages are improved learning capabilities, fast training, robustness, reduced computational burden as well as intuitiveness. A strategy combining an intelligent adaptive neural controller with a robust compensator is proposed by (Yu, 2014). While the robust compensation control law accommodates external disturbances, the switching neural controller approximates uncertain system-internal parameters. Through a Lyapunov-based switching adaptive law, the latter allows the tracking of any bounded output function with the overall controller's stability being guaranteed. An intelligent control-extension for SMC based on neural networks was presented by (Van Cuong, 2016). Reinforcement learning was applied to robot control by (Peters, 2008) and (Hester, 2011). The main limitations of this learning algorithm are the restriction to rather low-dimensional problems due to the nature of reinforcement learning itself and the complexity of memory, computation and sample (Strehl, 2006). The need for an a priori specified reward function and usually required large number of samples are both time- and resource-consuming and therefore limit the application areas of reinforcement learning (Barto, 2019) (Deisenroth, 2015) (Duan, 2017) (Hester, 2011) (Koenig, 2017) (Ng, 2000). Though reinforcement learning is a promising approach for facing the need for adaptable robot systems in heterogeneous applications and changing milieus, relevant applications showing efficiency and reliability are lacking in literature and industry (Polydoros, 2017) (Riedmiller, 2018). The development of robust and easy to implement reinforcement learning algorithms remains a challenge in the field (Barto, 2019).

With the aim to improve bio-inspired learning, (Lucas, 2004) developed a controller based on the emotional learning behaviour of the mammalian brain: BELBIC - Brain Emotional Learning Based Intelligent Control. The source of inspiration for this controller was the emotional learning behaviour of the Amygdala-Orbitofrontal system of the human brain and more precisely, Moren and Balkenius' computational model of the abstracted human limbic system comprised of the amygdala and the orbitofrontal cortex (Balkenius 2001) (Morén, 2000). Their work was inspired by the natural interplay of actuating amygdala and inhibiting orbitofrontal cortex. The core idea is to attach emotions to inputs and their corresponding outputs. A reward function, i.e., a positive emotion is attempted to be maximized through adaptation of current or future behaviour based on prior experiences. This is the innate behaviour which is observed throughout mammalians. The key advantages of this control strategy are improved learning capabilities, robustness and an intuitive algorithm. Figure 8 illustrates the working principle of the BELBIC-concept. In the network like structure of the amygdalo-orbitofrontal system, the amygdala, the actuator, maps sensory stimuli to their linked emotional responses while the orbitofrontal cortex, the preventer, inhibits connections as a response to changing aspirations or environment. The emotional stimulus-response's capacity to assist in making fast decisions is exploited in the framework of two-process learning and signal processing resulting in an action-generator (one output) based on sensory inputs and emotional cues (multiple inputs).

The BELBIC-control concept was applied to areas as diverse as induction motors (Daryabeigi, 2014) and mobile robotics (Sharbafi, 2010). (Yi, 2015) combined robust sliding mode control with an intelligent control element comprising an actuator and a preventer inspired on the mammalian limbic system.

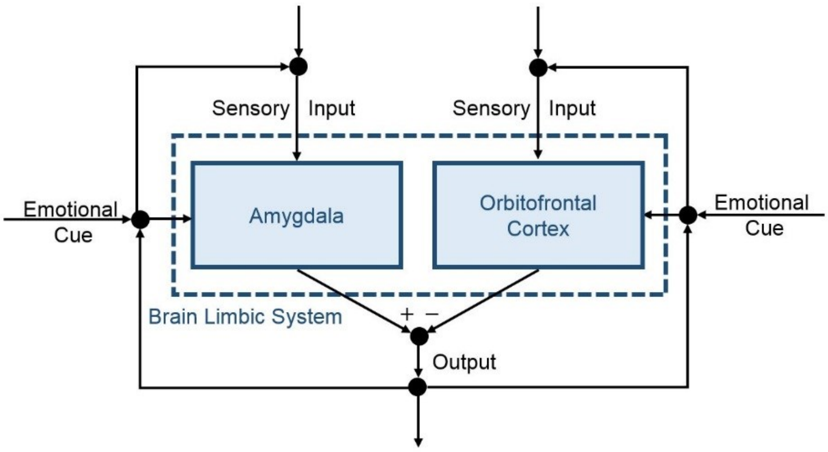



Figure 8: The principle of BELBIC – Brain Emotional Learning Based Intelligent Control.

The learning behaviour of the human brain inspired further developments in control engineering. Due to the complex nature of this biological system only a concise selection of its key-aspects has been retained for the development of control concepts. To learn action-outcome relations and achieve goal-based recall of these actions, (Baldassarre, 2013) presented a system-level model. The authors incorporated processes related to intrinsic motivations, i.e. motivations not directly stimulated by extrinsic rewards as well as to the role of repetition bias, i.e. the natural tendency to repeat recent actions. (Frank, 2014)'s work focussed on reinforcement learning allowing an agent to learn a policy with the goal to maximize a reward-signal. The authors combined a low-level, reactive controller with a high-level curious agent. Artificial curiosity contributes to the learning process by guiding exploration to areas where the agent can efficiently learn. The work was validated by a real-time motion planning task on a humanoid robot. (Merrick, 2012) implemented a goal-lifecycle and introspection for reinforcement learning. The aim was to make the system aware of when to learn what as well as of which acquired skills to keep either active, ignored or erased. (Franchi, 2016) simulated the structure and interaction among amygdala, cortex and thalamus, i.e. three parts of the brain which play a key role in human cognitive development. In the field of developmental robotics, the authors designed a network for autonomous learning with the ability to generate new goals. The suggested method was validated on an active sensing activity.

A variety of bio- and neuro-inspired features have been explored. These have neither all been structured as learning elements nor been implemented in control algorithms. The further development of these elements, their incorporation as learning extensions in robust control algorithms and the validation of deduced concepts for complex industrial applications in varying environments via simulation and experiments cannot be found in literature. Their combination is promising for addressing the remaining challenges in the field (Barto, 2019).

## 2.3 State-of-the-art: Estimation of Unknown Robot Parameters

As far as trajectory tracking control is concerned, most of the suggested algorithms assume precise knowledge of the robot dynamics. Their use is restricted to robotic manipulators with well-known and measurable dynamic and kinematic parameters. In practice, however, this is hardly ever the case, particularly in use cases which involve direct interaction between the robot end-effector and its environment (Sünderhauf, 2018). For a successful practical implementation of the theoretically developed algorithm, robot dynamics need to be estimated.

Different ideas to deal with uncertain, even unknown robot kinematics and dynamics can be found in the scientific literature. (da Silva, 2009) suggested a CA integrated design-framework for dealing with the control of machines with varying dynamics. (Ghalyan, 2016) (Jasim, 2014) worked on relaxing the need for knowing the precise, accurate robot dynamics through the use of fuzzy systems theory. In 1999, (Sun, 1999) suggested an adaptive fuzzy sliding mode control for trajectory tracking of robot manipulators with unknown nonlinear dynamics, combining the advantages of and overcoming the drawbacks of both fuzzy systems and sliding mode control. The fuzzy system was used here as an adaptive approximator for the robot's nonlinear dynamics. In 1997, the authors had developed another methodology (Sun, 1997): A neural network-based adaptive control concept for the same use case. In 2000, they presented an extension to their work (Sun, 2000), namely an estimation algorithm for the bounds on Neural Network approximation errors permitting the exclusion of the previously restricting bound assumption.

Because of their function estimation- and learning-capacities and their real-time compatibility due to low computational costs, neural networks are predominantly implemented (Liu 2013). I.a. in (Jung, 2000). The authors showed a force tracking impedance controller extended with a single neural compensator able to compensate for uncertainties in robot dynamics as well as in environment position and stiffness. The chosen two-layer feedforward Neural Network was trained with specific signals for free space and contact motion respectively. (Li, 2013) (Lin, 1996) (Xu, 2014) used neural networks to compensate for unknown dynamics. Feedforward neural networks were the base of the works by (Kumar, 2011) (Singh, 2013). (Ge, 1997) established his work on control of robotic manipulators with unknown dynamics also on Neural Networks. Through mathematical developments and numerical simulations, the authors showed that their algorithm based on Gaussian Radial Basis Function (RBF) networks achieved uniformly stable adaptation and asymptotical tracking without the need for a priori knowledge of robot dynamics or extensive, time-consuming training processes.

Due to their universal approximation-capabilities, fast convergence, straightforward training and analysis due to a simple topology and easy implementation due to few interconnections, RBF-NNs are willingly used (Bass, 1994) (Park, 1991) (Tao, 2016) (Van Cuong, 2016). RBF-NNs were applied to approximate nonlinear unknown robot dynamics (Lee, 2004) (Yu, 2014).

Theoretical developments of advanced robot model approximation concepts are abundant in scientific literature. However, further research is required for real-life validated estimation concepts. The interest lies in the simplest approximator not requiring training and guaranteeing satisfactory performance.

The choice of the network parameters, i.e. the node-centres and widths of the function is essential for the accurate performance of the network (Lewis, 1995) (Lewis, 1996). The RBF-NN is locally responsive, i.e. inputs which are close to the centre in the Euclidean sense strongly affect this node but not the others. The centres should be well selected according to the scope of the inputs, i.e. the values of the centres are to be suitably fixed and appropriately distributed in the input domain. The selected width influences the range over which a node is to have a significant activation. In the vast majority of research works, a priori selected centres and widths are kept fixed. Although it simplifies the analysis in dynamic systems, this approach presents some drawbacks (Bass, 1994) (Tao, 2016) (Wang, 2009). First, fixed parameters and therewith the mapping behaviour of the network do not reflect changes in the system which leads to suboptimal performance in dynamically changing environments. Because the parameters of the system under consideration may change over time, the neural network should be adapted online (Bass, 1994). Second, the initialization of the parameters is not straightforward. The arbitrary selection of values often practiced does not guarantee satisfactory performance. With well-fixed centres, a performance similar to the one of multi-layered networks can be expected. The determination of these centres, in practice chosen from data points, however causes problems due to the excessive amount of data to process (Chen, 1990). The determination of parameters through trial-and-error is also practiced (Xu, 2014). Real-time computation of parameters based on observed data could be extended from weights to function parameters (Van Cuong, 2016). The remaining challenge consists in reducing the dependence on a priori initialized and fixed network parameters.

Together with parameter initialization, stability is a challenge during the implementation of neural networks. Several research works addressed this issue through combinations of neural networks and robustifying elements. Using sliding mode control to stabilize a neural network was suggested. The downsides were the occurrence of chattering, i.e. undesired oscillations, high control efforts and the need for a priori knowledge of system limits (Bass, 1994) (Van Cuong, 2016). As an extension, (Ren, 2007) suggested merging a neural network with a combination of a PD- and a sliding mode controller (SMC). The aim of the PD and the SMC was to first bring the system's output into the targeted regions and to second provide robustness. The multilayer feedforward NN with input modification was used to approximate the system's desired output and improve its global performance. A stabilizing extension for neural networks was applied to robot manipulators with unknown dynamics (He, 2018) (Sun, 2011). A two-fold control concept was also suggested by (Lee, 2004) and (Singh, 2013): a function approximator based on Neural Networks for estimating unknown robot dynamics was combined with an auxiliary controller for compensating for uncertainties and approximation errors.

Further research is required on the merging of robust control algorithms and model approximators for a global stable and efficient system performance.

## 2.4 State-of-the-art: Combined Position and Force Control

Most manufacturing tasks require the robotic end-effector to interact with its environment which results in contact situations and constrained movements, i.e. the robotic manipulator cannot move freely in all directions. Constraints include natural constraints due to the specificities of the environment as well as artificial constraints due to and characteristic of the desired task. Varying or switching constraints are due to successive discrete or continuous phases in a task. By their nature, the control of constrained tasks requires the simultaneous control of pose and wrench (Phillips, 2016). Pure position control cannot cope with these complex contact-rich tasks because already slight deviations from the desired trajectory can lead to errors in the desired forces and torques (Abu-Dakka, 2015) (Kronander, 2014). In addition to the position, the applied contact force has to be controlled (Hazara, 2017). When the application of specific forces is required, stiff position controllers are insufficient (Siciliano, 2008). In this sense, contact-rich manipulation tasks that involve interactions with the environment are especially challenging (Englert, 2018) (Fu, 2016) (Kramberger, 2017) (Kronander, 2014). Pure force control on the other hand can lead to contact instabilities at an increased speed (Newman, 1999). This is the explanation for the interest in combined force- and position-control. Related to this control problem, two areas of concern arise. The first concern deals with the appropriate concept or algorithm for the combined control of force- and position-signals. The second question is how to obtain the desired reference position- and force-signals, i.e. the controller inputs.

Combined position- force-control and its extensions have been a topic of interest in the control community for the past decades. (Pagilla, 2001) implemented an event-based control switching strategy for a surface finishing process. The different process phases are identified and tackled with a respectively adapted control algorithm. In the 1990s, hybrid position- force-control (You, 1996) and its application to redundant robots in constrained tasks (Jankowski, 1999) were presented. For interaction tasks with rigid surfaces in the presence of uncertainties, an extension was suggested by (Pliego-Jiménez, 2015). Based on local estimations of constraint parameters and force measurements, the constraint surface was approximated and the trajectory modified online. For contact tasks with compliant planar surfaces, a passivity-based controller was developed (Siciliano, 1996). (Park, 2008) combined auto surface tracking algorithms and active compliant motion control to follow a surface and control the contact force between end-effector and workpiece. Motivated by the concern about robustness, (Ha, 2004) extended a hybrid position- force-controller with an adaptive scheme to estimate uncertainty-bounds. (Kiguchi, 2000) suggested a fuzzy neural position-force-controller. The algorithm was applied to a grinding task executed by a 2DOF planar robot while neglecting the approach phase. The transition from non-contact to contact situation was often omitted in control development processes (Siciliano, 1996). The separation of position- and force-controllers has been described i.a. in (Von Wattenwyl, 2001) where a decoupling in independent position- and force-control-loops was suggested. A hybrid algorithm comprising a closed-loop position- and an open-loop force-controller was applied to a grinding application (Fazeli, 2012). In the normal direction, a switch between position- and force-control was implemented. A neural network based hybrid position- force-controller with separate NN compensators along both force and motion controlled directions was suggested by (Ziauddin, 1994). (Moghadam, 2014) (Moghadam, 2014b) developed a hierarchical optimal position- force-control algorithm for contour tracking by CNC-machines.

One of the most promising approaches for the simultaneous control of force- and position-signals in this context is parallel control, i.e. a position- and a force-controller acting in parallel, if applicable with a priority strategy. Both independent controllers yield control torque commands which are summed up. Parallel control has been the subject of repeated research efforts over the past decades. Based on the interactions of controller, robot arm and environment, (Chiaverini, 1993) developed a dynamic parallel force- position-control for constrained motions with an elastic environment. (Ferguene, 2009) extended a conventional parallel force- position-controller with a 3-layer feed-forward neural network to compensate for uncertain or varying robot dynamics and environments. The intended application areas ranged from curved surfaces to unknown environmental stiffness. (Karayiannidis, 2007) (Karayiannidis, 2010) suggested adaptive concepts for position- force-control in compliant and frictional contacts in the presence of uncertainties in models, end effector-orientation and environment. (Yin, 2012) based his tracking controller on a human analogy, i.e. on the human's approach to finger tracking in the absence of visual feedback. For the tracking of an unknown surface, the authors relied on the concepts of moving frames and vector-variations. (Lange, 2013) presented a parallel position-based force- torque-control scheme taking into account couplings between forces and torques, sensor and environment, constrained configurations, compliances in robot as well as the effects due to impact forces. An experimental validation completed the work.

The implementation of combined robust and bio-inspired control-elements as discussed in 2.2 is left to be investigated in parallel control for combined position and torque control.

In the area concerned with the acquisition of the controller inputs, two methods are predominant. Whereas the first one relies on theoretical knowledge in abstract form, the second one aims to translate human expertise in a robot program.

In the case of non-parametrized trajectories, the first approach makes use of knowledge extracted from CAD/CAM-files. The idea is to provide the robot with the geometry to be followed, avoiding the exclusive reliance on sensory information. Information from CAD/CAM-files being only readily useable by CNC-machines, it has to be translated in order to be readable and useable by a robotic manipulator. A variety of commercially available software deals with this translation (KUKA, 2017) (Octopuz) (Robotmaster). The technology package KUKA CAM.Rob allows the machining by robots based upon data created from conventional CAD-files. Practical applications are mainly found in milling, coating and polishing of shaped workpieces. This method was developed and verified only for the big industrial KUKA KR robots and for highly specific applications. The use of similar technology packages is therefore restricted to individual robots and use cases and not applicable to the majority of automation tasks. This method can only be applied for workpieces with known geometries. These have to be available in CAD-format. Additionally, the closeness of these commercial solutions restricts their potential for research and further development.

The objective of the second approach is to mimic a human operator's behaviour. Behavioural cloning is a type of imitation learning, where the idea is to use supervised learning in a straightforward manner to learn from demonstrations. The need for more advanced control algorithms came from the method's inability to adapt or to overcome problems arising from small disturbances or uncertainties (Arulkumaran, 2017). Traditional industrial robotic systems lack methods for intuitive transfer of human expertise. This translation is a challenging task (Perzylo, 2019). A combination with reinforcement learning was suggested (Hester, 2017). (Skubic, 2000) used a hybrid control model composed of a low-level force control and a higher-level discrete event control to teach robots assembly skills from human demonstration. By reacting to changes in force-based qualitative states rather than to absolute positional information, the suggested method is based on contact formations, i.e. force-based discrete states describing qualitatively the established contact. Arbitrary forces were tracked using iterative learning (Gams, 2014). (Zhang, 2003) transferred and adapted knowledge from one task to another in a human-like language of fuzzy control rules. (Khansari, 2016) extracted desired forces from data analysis. (Xu, 2007) computed forces via an algebraic function of states and constraints for a sensor-less position- and force-control. (Oba, 2016) discussed the acquisition and replication of polishing skills of a human worker represented as tool trajectory, tool posture and polishing force. These variables which were controlled independently and simultaneously formed the input to the controller. (Abu-Dakka, 2015) presented a concept for adaptive learning of contact-based manipulation tasks. The authors suggested a scheme for online modifications to match desired reference position- and force-profiles. The latter were obtained from programming by demonstration and encoded with dynamic movement primitives. (Yang, 2018) facilitated human-to-robot skill transfer through subdivision of global skills and subsequent regulation of subskills. The authors further included an estimation of human limb stiffness into their variable impedance control scheme of the robot for the skill transfer.

The most straightforward method to transfer know-how from human operators to robotic manipulators is by showing the desired behaviour. (Rozo, 2013) used Programming by Demonstration (PbD). Based on Gaussian mixture theory, a single model encompassed both, desired positions and forces. The interest of PbD and kinaesthetic teaching for automating complex manipulation tasks was also pointed out by (Koenig, 2017) (Siciliano, 2008). This supervised method enables robots to be programmed intuitively by non-experts. Intuitiveness is one of the key elements required for the future of flexible robotic applications in dynamic environments alongside humans (Villani, 2018). According to (Johnson, 2019), the highest potential of new technologies and especially of AI, consists in teaming intelligence, i.e. in fully integrating automation into human work practice instead of taking the human out of the loop and resorting to fully automated systems. The benefits of combining human and artificial intelligence were discussed from the points of view of state, structure, skills and strategy.

Programming by demonstration aims to leverage human expertise to teach robots via guided examples. (Koenig, 2017) combined learning from demonstration with life-long learning in a concept for continually learning and adapting in a robot programming framework. The work was based on influence diagrams. (Kormushev, 2011) used torque-based kinaesthetic teaching on lightweight robots as an intuitive and user-friendly method to transfer skills from human operators to robots. The authors considered both position and force aspects. (Kober, 2015) suggested a hybrid position-force-control scheme based on kinaesthetic teaching. (Zhang, 2018) developed a task-agnostic machine learning approach for robotic assembly tasks. Cartesian poses and wrenches which were recorded via kinaesthetic teaching formed the input for the system. The research of the Institute of Robotics and Mechatronics of the German Aerospace Center focussed on torque-controlled lightweight robots. One of their designs is the KUKA LWR showcasing a zero gravity mode. This mode allows the user to operate the robot arm naturally as if it had no weight (Albu-Schäffer, 2007). These developments were among others used in the works of (Khansari, 2016) (Kramberger, 2017) (Montebelli, 2015) (Rey, 2018) (Zhang, 2018). The subdivision of desired position- and force-profiles was a topic in various research works. (Kormushev, 2011) addressed first the desired position- and then the force-profiles in an ironing application requiring varying stiffness. The subdivision of a manipulation task into kinematic- and force-driven phases was employed in (Lee, 2015). In the suggested control algorithm a trade-off between position- and force-error depending on the stage of the task was implemented. Based on demonstrations acquired via kinaesthetic guiding, (Kramberger, 2017) statistically generalized position-, orientation- and force-/torque-profiles to compute adapted control policies for contact-rich assembly tasks. The authors validated their method on an impedance controlled KUKA LWR4+-robot manipulator. (Akgun, 2016) focused on the humans' goal-oriented approach to improve a reinforcement learning based concept for learning from demonstration. The authors differentiated between learned actions and learned goals, i.e. rewards. (Kalakrishnan, 2011) divided desired position- and force-profiles. Here, the latter were addressed through reinforcement learning. A model-based reinforcement-learning algorithm combining prior experience encoded in a neural network dynamics model, online adaptation and planning based on model predictive control was applied to manipulation skills (Fu, 2016). (Kober, 2011) also made use of reinforcement learning to adapt to new situations trying to reduce the amount of required prior knowledge. (Hazara, 2017) used PbD and a Dynamic Movement Primitives extractor for an intuitive simultaneous recording of position and force signals. The authors combined this step with a Cartesian impedance controller and a reinforcement learning-based optimizer for addressing in-contact skills via control of position and normal force. Their concept was validated on a KUKA LWR4+ with additional force/torque sensor for planing wood boards. The major drawbacks of reinforcement learning are the need for an a priori specified reward function as well as the large number of samples which is usually required for a satisfying performance (Barto, 2019) (Deisenroth, 2015) (Duan, 2017) (Koenig, 2017) (Ng, 2000). The motivations for the work were the high costs of experience as their generation generally is time-consuming and requires the involvement of humans. A similar motivation was at the base of the works of (Domroes, 2013) (Duan, 2017) (Englert, 2018) (Martínez, 2017) (Navarro-Gonzalez, 2015) (Silver, 2017). (Englert, 2018) implemented reinforcement learning to allow a robot to perform a manipulation task with objectives and only a single demonstration as input. (Duan, 2017) developed a meta-learning framework based on NN to learn and control a task using only one or very few demonstrations. The concept of one-shot imitation learning was here applied to a family of block stacking tasks. (Martínez, 2017) suggested an algorithm based on rule analysis and relational reinforcement learning which asked for teacher demonstrations only when a significant improvement was expected.

There is a need for generating controller-inputs with kinaesthetic teaching ensuring the transfer of human know-how to the automated system. In this context, the compatibility of kinaesthetic teaching and robust adaptable control algorithms is essential as well as the simplification of the PbD-step.

## 2.5 Discussion and Concluding Summary

The present part reviews this chapter's previous parts, i.e. the literature surveys on position control, estimation of unknown robot models, combined position and force control with a focus on the identified shortcomings. From this, the scientific gaps or areas for further research are clearly elaborated and the link is laid to the third chapter.

For the trajectory tracking control problem in the case of known robot parameters, model-based sliding mode control has proven to be a prominent choice thanks to robustness and invariance to external and internal uncertainties. Mostly, control algorithms are validated on simple continuous path following tasks in ideal conditions. The discontinuous tracking of complex freeform surfaces results in the presence of switching constraints, including transitions from free space to contact situations, and impedes the robust performance of the suggested controllers. Uncertainties and disturbances affect the robustness and stability in real-life scenarios. In this context, they should be included in simulations and controllers validated in both simulation and real-life experiments. Controller-extensions have been suggested to address these issues and overcome the drawbacks of SMC: need for a priori knowledge of function bounds and chattering. Reinforcement learning is a promising approach. Validated freeform trajectory tracking in industrial applications, however, are missing. Bio-inspired algorithms are optimal due to their intuitiveness and because they take advantage from the natural evolution. A variety of bio- and neuro-inspired features have been explored by the scientific community. These have neither all been structured as learning elements nor been implemented in control algorithms. The further development of these elements, their incorporation as learning extensions in robust control algorithms and the validation of deduced concepts for complex industrial applications in varying environments via simulation and experiments cannot be found in literature. Combining sliding mode control and bio-inspired learning controller-extension is suggested for an advanced trajectory tracking application in the presence of switching constraints and uncertainties. An advanced neuro-inspired reinforcement learning algorithm is to overcome the excessive need for samples and memory-costs.

Uncertain or unknown robot models make a straightforward implementation of model-based control algorithms impossible. For the necessary approximation of the unknown nonlinear robot dynamics, theoretical developments of advanced concepts are abundant in scientific literature. I.a. artificial neural networks have been widely investigated. Although the mentioned publications showed promising results for robotic control, the results have mainly exclusively been obtained in numerical simulations. The majority of simulations have been performed for n-link-robot arms with n=2. However, for n>2, model-parameters affect the performance of the system more considerably and their initialization becomes even more critical (Krabbes, 1999) (Yu, 2014). Further research is therefore required for real-life validated estimation concepts. There is an interest in the simplest approximator not requiring training and guaranteeing satisfactory performance. The main shortcoming of NNs is the need for its parameters to be appropriately predefined and initialized. Real-time computation and adaptation of parameters based on observed data was extended from weights to function parameters (Van Cuong, 2016). To overcome this deficiency the estimation of unknown robot dynamics parameters through a RBF-NN with adaptive parameters is suggested in combination with robust control elements. Further research is required on the interlocking of robust model-based control algorithms and model approximators for a globally stable and efficient system performance.

Programming by Demonstration and kinaesthetic teaching are prominent methods for skill transfer between humans and robots due to their intuitiveness and ease of application. Recording desired signals with kinaesthetic teaching ensures the transfer of human know-how to the automated system. PbD is preferred to the more conventional method of generating input-signals from CAD/CAM-files due to the limited range of applications of the latter. The use of kinaesthetic teaching is combined with adaptive control algorithms to provide a stable, adaptive and intuitive automation. The compatibility of kinaesthetic teaching and robust adaptable control algorithms is essential as well as the simplification of the PbD-step. Combining PbD and sliding mode control elements with extended neuro-inspired reinforcement learning elements in a parallel control setting is intended to overcome the shortcomings identified in existing control-approaches. The areas of application of robotic manipulators has to be extended to complex multiaxial in-contact tasks.

### 2.5.1 Scientific Gap

A lack in experimentally proven control algorithms for position and combined position-torque- control for complex trajectory tracking applications has been identified. Discontinuities, switching constraints and uncertainties have only dissatisfyingly been addressed. Performance in terms of reduced average error signals over the total process time has not been focussed on. The limiting factors of the estimation of potentially unknown models through Neural Networks were their fixed parameters. Robustness, adaptive bio-inspired learning and human expertise have only separately been addressed. Validations beyond numerical simulations on simple robots are desirable.

# 3 Research Question

The purpose of the third chapter is to explain how the identified scientific gaps are going to be filled. This is done in three phases, each answering a specific question (Figure 9).

The research questions are directly deduced from the gaps identified in 2.5.1. This section answers the question what is researched in the present work.

The scientific hypothesis answers the question how, with which concepts the research questions are addressed.

In the third phase, the question is: how or by what means will the hypothesis be confirmed?

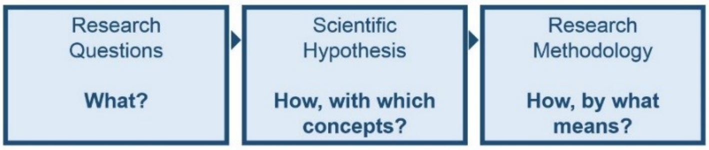



Figure 9: Structure of chapter 3 Research Question.

<u>Research Questions</u>

- Can freeform trajectories (successions of straight, convex and concave segments) in the presence of switching constraints and uncertainties (transitions from free space to in-contact-situations and external disturbances) be tracked with a performance of reduced absolute mean position error (<0.3 rad)?
- How can robustness be combined with adaptability?
- How can the dependence on a priori knowledge of robot model parameters be eliminated?
- Can desired forces be applied at specified positions on the tracked trajectory? Performance evaluation through reduced absolute mean torque error (<0.1 Nm).
- How can be taken advantage of human expertise?

<u>Scientific Hypothesis</u>

This work investigates the following hypothesis:

A position and force/torque control algorithm combining robustness characteristic of conventional robot-control with adaptability typical for humans is able to answer the research questions related to the control problem of complex trajectory tracking in adverse conditions.

<u>Research Methodology</u>

The question answered in this part is: ‘how, by what means are the solutions to the research questions to be found and the scientific hypothesis to be confirmed?’. The main aspects of the research methodology can be grouped and detailed as follows:

- Synergizing classical control concepts appreciated for their robustness with bio-inspired learning algorithms known for their adaptability allows the fusion of the concepts’ respective advantages. Complex trajectory tracking can be addressed via
  1) a combination of SMC and BELBIC,
  2) a combination of SMC and an extension based on reinforcement learning.

- The control algorithms are to be invariant to unknown robot models. This is to be achieved either through model-free algorithms or through approximating the robot dynamic parameters in model-based concepts. This is done
  1) as a single nonlinear robot function to reduce the estimation burden and the impact of approximation errors,
  2) using a simple perceptron NN-model for the ease of design and implementation,
  3) with online update-laws for the NN-parameters to overcome the drawback of a priori fixed parameters.
  Stability and performance (absolute mean position error <0.3 rad and torque error <0.1 Nm) are not affected by the lack of knowledge about the robot model.
- Combined position and force/torque control can optimally be addressed via
  1) a parallel concept with independent position and torque controller-halves,
  2) each controller-half combining robust control with an adaptive reinforcement-based learning element.
- Human expertise can be included through
  1) following the human operators' bottom-up approach of subdividing the comprehensive control challenge into a) trajectory tracking and b) trajectory tracking with force/torque application,
  2) using desired position- and torque-signals originating from kinaesthetic teaching as input for the control concept.
- In order to assess the stability and performance (absolute mean position error <0.3 rad and torque error <0.1 Nm) of the developed control concepts beyond theory and simulation, real-world experiments are performed.

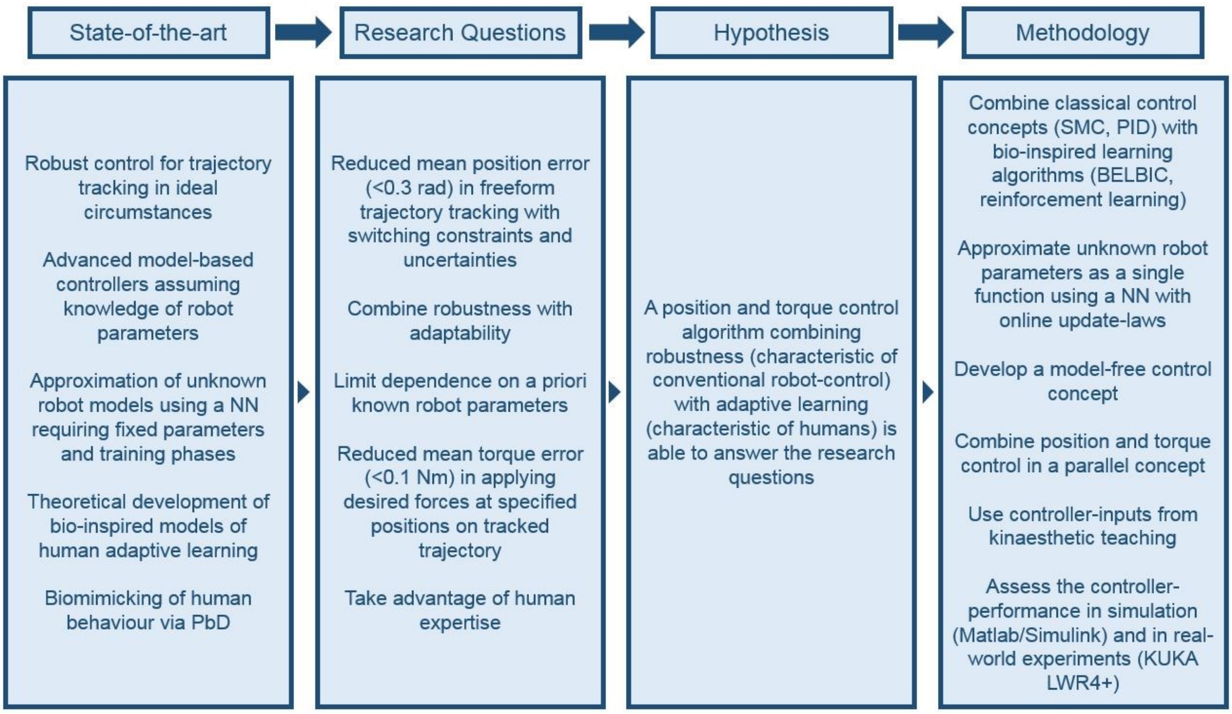



Figure 10: Overview of State-of-the-art, Research Questions, Hypothesis and Methodology.

# 4 Two-Steps-Approach for Performance-Evaluation

A theoretical derivation and stability analysis are first steps in the development of a novel control algorithm. As assessment of the performance of the suggested algorithm either simulations or lab-tests can be performed. Because both strategies bear advantages and disadvantages, the presented setup for validation combines both.

The implemented performance-evaluation strategy follows a two-steps-approach.

The first step is a numerical simulation. The simulation-software Matlab/Simulink is chosen because of its widespread use in academia and industry and because of its features (i.a. wide design space, combination of textual and graphical programming, matrix manipulation and graph plotting features). Although a simulation cannot offer all the elements encountered in a real-life setting it gives a good indication of the performance of a controller. The simulation is performed on one of the simplest forms of robotic manipulators: a 2D planar RR-robot arm. Although the goal is to implement the control algorithms on a much more complex robotic manipulator, the simple RR-arm contains most elements of the more advanced manipulators. A simulation of a 2-link robot allows to conclude whether the validation steps should be pursued.

In the second step, the control-algorithms are implemented on a robot in the university-laboratory. The experimental evaluation is performed on a KUKA LWR4+. Due to its size and morphology which resembles that of a human arm, this manipulator is well suited for replacing the human operator and its relatively small payload is no issue for the intended surface finishing processes. The rather open architecture and the FRI allow the implementation of own controllers in a widespread programming language. Further, kinaesthetic teaching is enabled by the features of the manipulator. The objective is to work without additional sensors.

This chapter describes both setups which will then be used in chapters 5 and 6.

## 4.1 Simulation Setup

As performance-assessment of the suggested control scheme, the controller will be verified through simulation using the Matlab/Simulink-environment. For the simulation a two-link planar robotic manipulator with revolute joints as shown in Figure 11 is used.

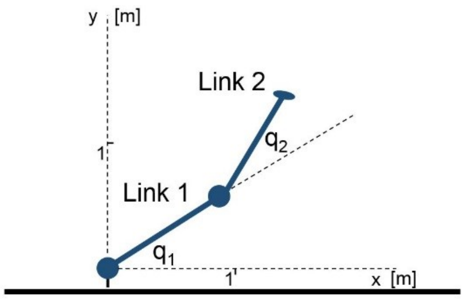



Figure 11: 2-link planar RR-robot arm as considered in the simulations.

For this manipulator the inertia, centripetal/Coriolis and gravity terms are given in eq. (5), (6) and (7), respectively (Shah, 2009).

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11} = j_1 + m_2 {l_1}^2$$

$$M_{12} = m_2 l_1 l_{c2} c_{21}$$

$$M_{21} = m_2 l_1 l_{c2} c_{21}$$

$$M_{22} = j_2 + m_2 {l_{c2}}^2 \tag{5}$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$C_{11} = 0$$

$$C_{12} = -m_2 l_1 l_{c2} s_{21}$$

$$C_{21} = m_2 l_1 l_{c2} s_{21}$$

$$C_{22} = 0 \tag{6}$$

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

$$G_1 = m_1 g l_{c1} c_1 + m_2 g l_1 c_1$$

$$G_2 = -m_2 l_{c2} g c_2 \tag{7}$$

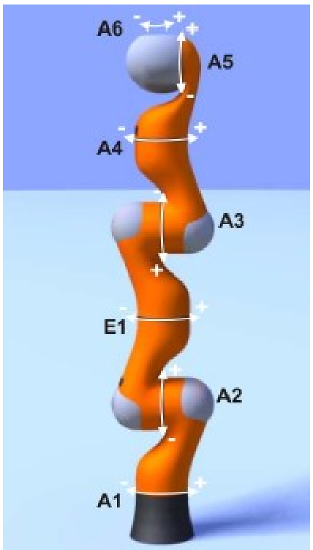
with $m_1, m_2 = 1kg$ and $l_1, l_2 = 1m$ the masses and lengths of links 1 and 2, $g = 9.8m/s^2$ the gravitational acceleration. $l_{c1}$ and $l_{c2}$ are the distances of the respective link's centre of mass from the source end. $j_1$ and $j_2$ are the moments of inertia of both links about their respective centres of mass. Further, $s_1 = \sin(q_1), s_2 = \sin(q_2), c_1 = \cos(q_1), c_2 = \cos(q_2), s_{21} = \sin(q_2 - q_1), c_{21} = \cos(q_2 - q_1)$.

## 4.2 Experimental Setup

As a demonstration of the performance of the suggested controller in real-world applications beyond numerical simulation, it will be implemented on a KUKA Lightweight Robot (LWR), a 7DOF- KUKA LWR4+ (Bischoff, 2010). Bio-inspired by the human arm and with a payload of 7 kg and its 7 axes, all equipped with internal position- as well as force-/torque-sensors, this redundant robot offers a range of features which are essential for the considered application. The seven revolute joints of the industrial robot are driven by brushless motors via harmonic drives. The specificities are indicated in Table 1. The working envelope is described in Figure 12. By default, the KUKA LWR 4+-robot is programmed via its attached controller and teach pendant in KRL-KUKA Robot Language (Figure 13 (left)). The implementation and validation of the suggested control scheme requires a more flexible and advanced programming environment. The LWR's possibility to connect to and communicate with an external PC through the FRI (Fast Research Interface) is used (Figure 13 (right)). The latter is a software-option provided by KUKA for experimental work in research-laboratories. The software configures a UDP socket communication for a binary data transfer with a cyclic timeframe in the range of 1 to 100 ms. The use of UDP socket communication is well suited for this use case due to its speed. It allows data exchange to and from the external computer, e.g. reading and writing: measurements from the robot-sensors are read by the PC and commands programmed in C++-language are sent from the PC to the robot (Figure 14) (KUKA, 2011) (Schreiber, 2010). Figure 15 shows the working principle of the FRI. The communication is ensured through two different modi: monitor and command mode. Through the order 'FRIOPEN' monitor mode is started where the robot can only be observed by the external PC, i.e. sensor measurements can be read. Through the order 'FRISTART' and only when the communication quality is sufficient command mode can be started. In this mode an interaction with KRL-programs as well as the execution of customer programs written in C++-language from the external PC are possible. This can be done in cartesian impedance, joint impedance or joint position control modes.

Table 1: Specifications of the KUKA LWR4+.

| | |
|---|---|
| **Controller** | KR C2 lr |
| **Mounting flange A 6** | DIN ISO 9409-1-A 50 |
| **Mounting position** | Any |
| **Number of axes** | 7 |
| **Payload** | 7 kg |
| **Repeatability (ISO 9283)** | ± 0.05 mm |
| **Weight (- controller), approximately** | 16 kg |
| **Wrist variant** | In-line wrist |




| Axis data | Motion range | Speed with rated payload | Maximum torque |
|---|---|---|---|
| Axis A1 (J1) | ± 170° | 110°/s | 176 Nm |
| Axis A2 (J2) | ± 120° | 110°/s | 176 Nm |
| Axis E1 (J3) | ± 170° | 128°/s | 100 Nm |
| Axis A3 (J4) | ± 120° | 128°/s | 100 Nm |
| Axis A4 (J5) | ± 170° | 204°/s | 100 Nm |
| Axis A5 (J6) | ± 120° | 184°/s | 38 Nm |
| Axis A6 (J7) | ± 170° | 184°/s | 38 Nm |

Figure 12: Axes-nomination of the KUKA LWR4+ (KUKA, 2012) (left); Description of the work envelope of the KUKA LWR4+ (right).

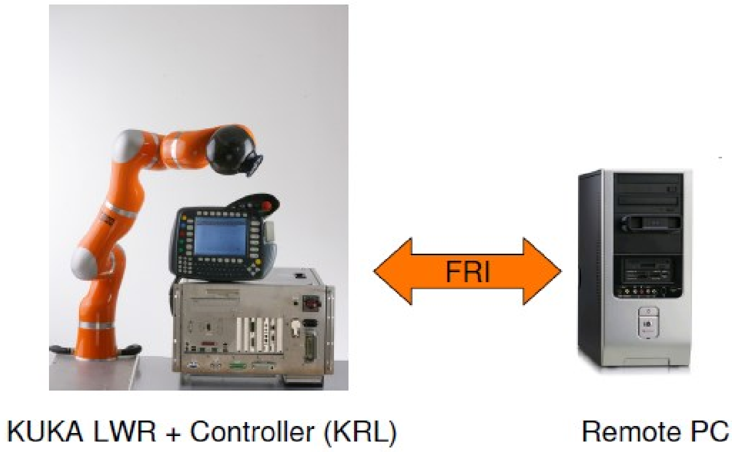



Figure 13: Programming of the KUKA LWR4+ through the teach pendant (left); Or through an external PC via the FRI-Fast Research Interface (right) (Schreiber, 2010).

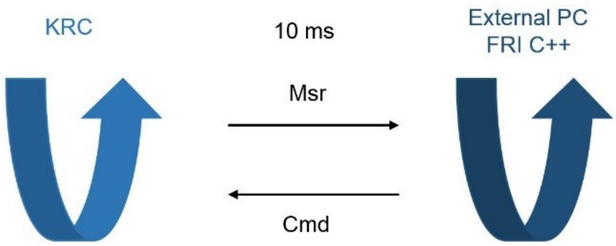



Figure 14: Data exchange between the KUKA LWR4+ and the external PC.

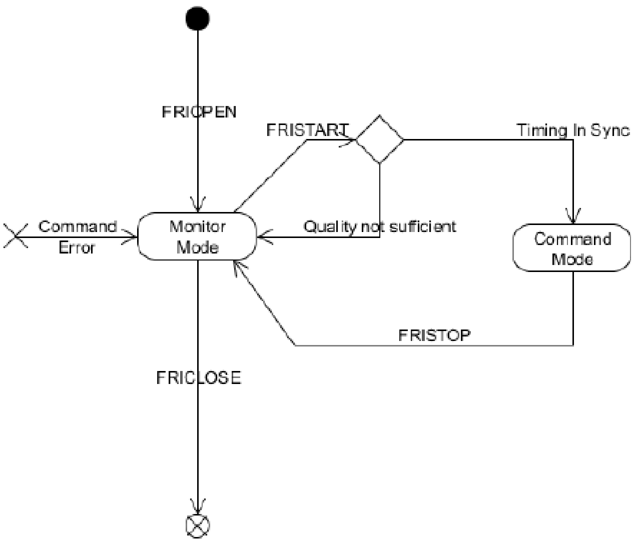



Figure 15: Working principle of the FRI-Fast Research Interface (Schreiber, 2010).

The KUKA LWR4+ is equipped with a zero gravity and gravity compensation mode (Albu-Schäffer, 2007). This mode allows a human operator to move the robot arm as if it had no weight and is specifically beneficial in a Programming by Demonstration-framework via kinaesthetic teaching. Torque-based kinaesthetic teaching is a user-friendly programming method for the KUKA lightweight robot. In contrast to tele-operation or hard-coded programming, two of the most common robot programming alternatives, kinaesthetic teaching is more intuitive as it allows the non-expert user to obtain direct feedback of the robot's body limitations and hence guarantees a certain first-hand perception of the task (Koenig, 2017) (Kormushev, 2011). The use of a robot arm like the KUKA LWR4+ allows the application of these methods without the use of additional sensors (Kramberger, 2017).

The features of the KUKA lightweight robot manipulator described in this chapter are used during the experimental validation of the developed control concepts. Figure 16 shows the test stand including a KUKA LWR4+ with a specifically designed and manufactured end-effector/tool and a metal workpiece. The freeform workpiece comprises a succession of concave, convex and straight surfaces. This test stand was built in the university-laboratory for experimental validation of control algorithms for automated constrained manufacturing tasks. Figure 17 shows the kinaesthetic teaching process. All experiments were performed in the 'Joint Impedance' control mode.

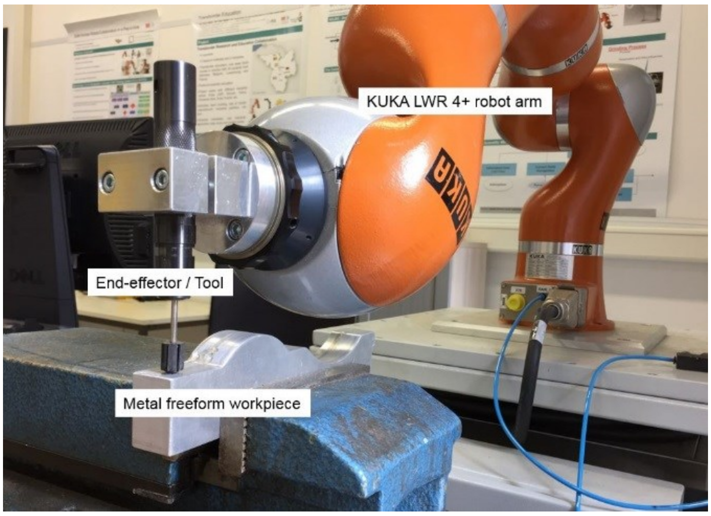



Figure 16: Test stand with the KUKA LWR4+ performing a trajectory tracking operation.

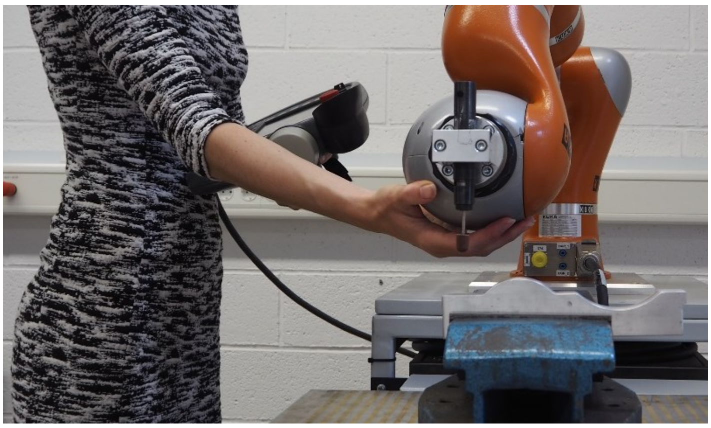


Figure 17: Kinaesthetic teaching.

# 5 Position Control

The fifth chapter is dedicated to the first sub-challenge which was identified in 1.1.4, namely path following, position control. The tracking of complex freeform-trajectories by robotic manipulators is essential to many manufacturing processes like grinding, welding, polishing or gluing. Besides pick-and-place operations, path following is the most common type of automation tasks (Giusti, 2018). With conventional controllers satisfactory performance is obtained for basic constrained motions and therefore they are still widely used in industry. The use of these conventional control schemes is however restricted to robotic manipulators with well-known dynamic and kinematic parameters following rather simple continuous paths in a disturbance-free environment. The desire to extend position control to robotic manipulators with unknown parameters following discontinuous freeform paths in the presence of disturbances explains the interest in the trajectory tracking control problem.

The purposes of this chapter are first to define the addressed position control problem with the requirements for its solution and second to develop and validate appropriate control algorithms. The structure of the chapter is illustrated in Figure 19.

The first part, 5.1, describes the position control challenge. First, the problem is formulated. Second, the requirements are deduced from the research question (chapter 3), the application area (chapter 1) and the gaps identified in the state-of-the-art (2.5). Third, the modus operandi is introduced.

The subsequent parts, 5.2, 5.3 and 5.4 present the suggested control concepts using the established modus operandi. 5.2 develops the model-based BELBIC-SMC algorithm. 5.3 suggests the BELBIC-SMC-NN algorithm comprising a neural network-based extension for estimating unknown robot model parameters. In 5.4, an extended algorithm is suggested which eases the need for fixed initialized variables and thereby facilitates the implementation. The iterative bottom-up procedure of parts 5.2 to 5.4 is twofold. Starting from the least complex use case in 5.2, difficulties impacting the performance of the control concepts are added step by step. In 5.3, the case of unknown robot model parameters is additionally addressed. In 5.4, the goal is to ease the need for fixed initialized variables thereby facilitating implementation on different systems and for different applications. Moreover, in ascending order, the suggested algorithms are further from concepts found in literature as presented in chapter 2.

In 5.5, the results of this chapter are summarized and discussed.

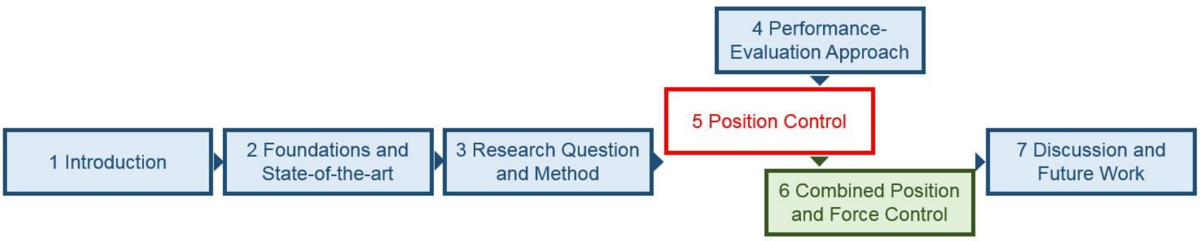



Figure 18: Situation of chapter 5 Position Control.

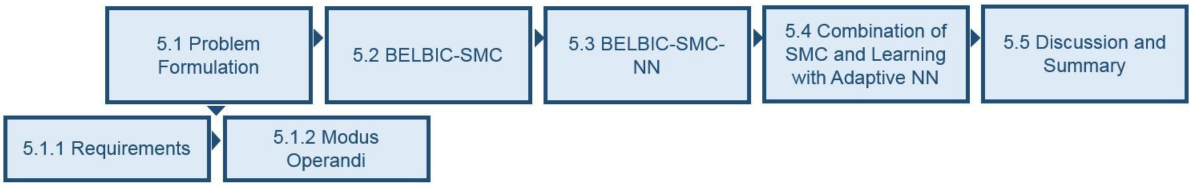



Figure 19: Structure of chapter 5 Position Control.

## 5.1 Problem Formulation

Conventional controllers exhibit satisfactory performance for a range of industrial use cases. Their use however is restricted to robotic manipulators with a priori known dynamic and kinematic parameters following continuous paths in a disturbance-free environment. Figure 20 recapitulates the circumstances which adversely impact the stable trajectory tracking of a system and therefore ask for more advanced control algorithms. The interactions between manipulator and surrounding environment as required by contact-based applications engender the former's movement to be constrained by the latter. Changes in constraints due to freeform geometries, i.e. successions of free space and contact situations, convex, concave and straight paths with a certain amount of uncertainty result in a switched nonlinear system. Further, robot-manufacturers' reticence concerning their products' kinematics and dynamics, uncertainties in the used models and external disturbances considerably affect the trajectory tracking performance. Therefore trajectory tracking control still requires some research in order to be applicable for more complex use cases.

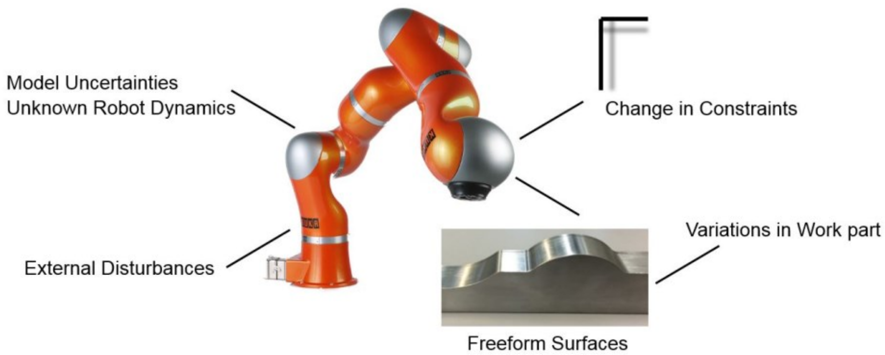



Figure 20: Effects impeding the overall stable system-performance in real world trajectory tracking applications (Touchlab UCL, 2014).

This thesis addresses the discontinuous freeform trajectory tracking control problem of robotic manipulators in the presence of switching constraints. The dynamics of the considered n-link robotic manipulator with switching constraints and disturbances can be expressed in Lagrange form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{Q}_i + \boldsymbol{d} \tag{8}$$

with $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in R^n$ link position, velocity and acceleration, $\boldsymbol{M}(\boldsymbol{q}) \in R^{nxn}$ the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in R^{nxn}$ the centripetal/Coriolis terms, $\boldsymbol{G}(\boldsymbol{q}) \in R^n$ the gravitational torque-vector,

$\boldsymbol{u} \in R^n$ the applied input torque. Interactions between manipulator and surrounding environment engender the former's movement to be constrained by the latter. The succession of discrete or continuous phases in a manufacturing process leads to a change in constraints between robot end-effector and environment. The transition from free space to contact situation is a common example. These varying constraints interfere to a more considerable extent with the overall system's stability than fixed constraints (Liberzon, 2003). The result is a switched nonlinear system. Here $\boldsymbol{Q}_i \in R^n$ is the global constraint force, $\boldsymbol{Q}_i = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{D}_i^T(\boldsymbol{\alpha})\boldsymbol{\lambda}$ where $\boldsymbol{J}(\boldsymbol{q}) \in R^{nx6}$ is the manipulator's Jacobian, $\boldsymbol{\lambda} \in R^m$ is the vector of Lagrange multipliers and $\boldsymbol{D}_i(\boldsymbol{\alpha}) = \frac{\delta \boldsymbol{\phi}_i(\alpha)}{\delta \alpha}$ is the gradient of the task space constraints with $\emptyset_i(\boldsymbol{\alpha}) \in R^6$ the i[th] kinematic constraint due to the system's environment. $\boldsymbol{\alpha} \in R^6$ stands for the Cartesian pose and $i = 1,2, \dots m$ denotes the index of constraints for the case of multiple switching constraints with $m$ the total number of constraints. A bounded external disturbance is introduced as $\boldsymbol{d} \in R^n$. $\boldsymbol{Q}_i$ and $\boldsymbol{d}$ are of opposite sign than $\boldsymbol{q}_{error}$. System-inherent uncertainties are included in the simulation as up to 10%-variations on the dynamic robot-parameters.

The introduced dynamics have the following two relevant properties (Slotine, 1991)

- $\boldsymbol{M}(\boldsymbol{q})$ is a positive definite matrix (P1),
- $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is a skew symmetric matrix, i.e. $\boldsymbol{x}^T(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))\boldsymbol{x} = \boldsymbol{0}$ for all $\boldsymbol{x} \in R^n$ and $\boldsymbol{x} \neq \boldsymbol{0}$ (P2).

The robot dynamics can be grouped and expressed as a nonlinear function

$$\boldsymbol{f} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G}(\boldsymbol{q}) \tag{9}$$

where $\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{F}\,\boldsymbol{q}_{\boldsymbol{error}}$ with $\boldsymbol{F} \in R^{nxn}$ is a constant, positive gain matrix.

Model-based control algorithms require knowledge of the controlled system. In the addressed trajectory tracking control problem, this concerns the robot dynamics i.e. $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ from eq. (8). In the case of partially or totally unknown systems, the parameters need to be estimated.

### 5.1.1 Requirements

The intended control algorithms address path following on freeform-workpieces with known, constant positioning. The aim is to develop bio-inspired algorithms for position control in an automated trajectory tracking process inspired by the human approach.

For the defined challenge of position control, i.e. trajectory tracking, the general requirement is the most optimal following of the desired trajectory: the diminution of the difference between actual and desired angular position, i.e. the minimization of the joint error. In more specific terms, the requirements involve the reduction of maximal, minimal and mean error signals with a special focus on the latter. Though it is desirable to keep the error in reasonable bounds, i.e. to have its minimal and maximal values as small as possible to prevent the end-effector from either taking off or penetrating the workpiece in a surface finishing process, special focus is on a constant satisfactory path following performance, i.e. minimization of the absolute mean position error below a threshold value of 0.3 rad. This value was chosen based on the results obtained with conventional controllers (PID and CJT) and on the performances described in the literature (Huang, 2003) (Jasim, 2015). The impact of interferences as described above is taken into consideration. The value is realistic and achievable for all manipulator joints, yet a significant improvement compared to the state-of-the-art.

The velocity of the trajectory tracking is only of minor relevance. Indeed, the focus is not put on the speed of execution, as also in humans it has been proven that precision and speed are negatively correlated, i.e. to increase precision of its action, the human slows down (Fitts, 1954).

For the case of unknown robot dynamic parameters, the aim is to approximate them in the shape of the nonlinear robot function $\boldsymbol{f}$ (eq. (9)). Satisfactory performance in terms of the position error requirements is needed when the estimated robot function is used. In other terms, the performance (absolute mean position error <0.3 rad) is not affected when using the approximated robot model parameters.

Following the vision of a human mimicking concept with facilitated implementation and use, compliance with PbD is desired. .csv-files acquired through kinaesthetic teaching are required to function as inputs for the controller.

### 5.1.2 Modus Operandi

5.2, 5.3 and 5.4, i.e. the parts dedicated to the suggested algorithms, each follow the outline depicted in Figure 21. After their theoretical development including the Lyapunov-stability analysis, the proposed algorithms are validated. First through simulation in the Matlab/Simulink-environment. Second through experimental validation on the lightweight robot arm KUKA LWR4+ in the laboratory. The respective results are shown. To conclude, they are analysed and summarized.

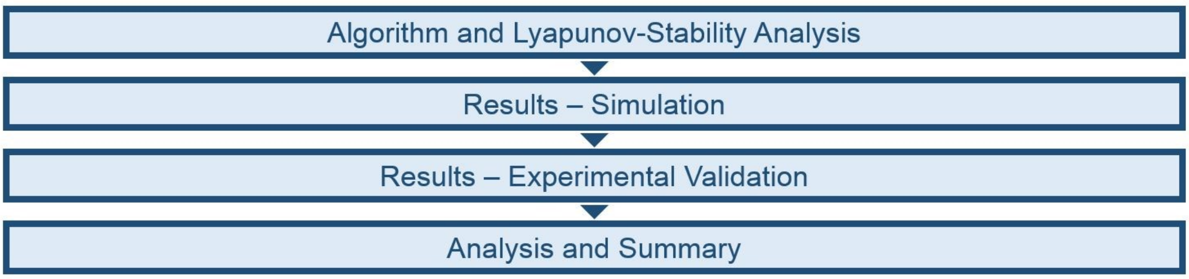



Figure 21: Modus Operandi.

## 5.2 BELBIC-SMC

From the literature survey, two essential topics in position control problems of robotic manipulators were identified: Robustness and adaptability or learning capacities. On the one hand, sliding mode control is ideal to address nonlinear systems as it is robust and invariant to system-internal and –external uncertainties. On the other hand, bio-inspired BELBIC, a reward-based learning concept is chosen for its fast learning abilities and adaptability to variations in the environment. Despite their excellence in their specific area of expertise, both control concepts present drawbacks. While SMC requires a priori knowledge of the system and is sensitive to imperfect control switching, BELBIC lacks the general robustness of the sliding mode controller. In this work, the combination of SMC and BELBIC control concepts is suggested in order to compensate the respective drawbacks and fuse their advantages.

### 5.2.1 Algorithm and Lyapunov-Stability Analysis

For the tracking control problem of discontinuous freeform-trajectories, BELBIC and SMC are combined. A conventional controller is used as starting point. In a controller extension, elements from sliding mode and from BELBIC-control are intertwined. The control-outputs of the conventional controller and the BELBIC-SMC-extension are added up. This addition of the extension to the conventional controller is illustrated in Figure 22.

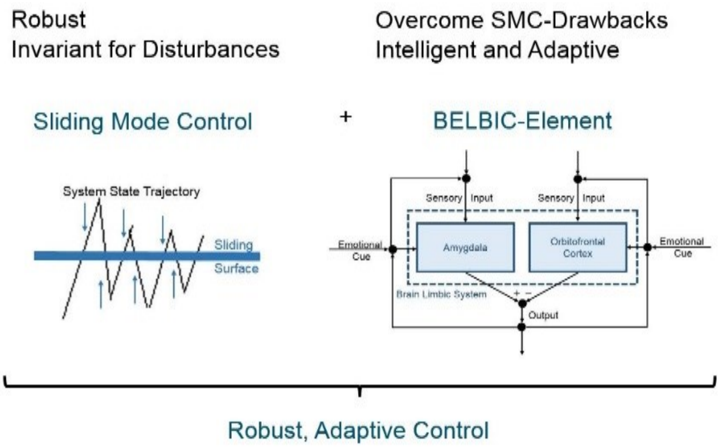



Figure 22: Combination of the conventional controller and the BELBIC-extension.

The suggested control concept (eq. (10)) corresponds to a controller structure with two parts, where $\boldsymbol{u}$ is the total input vector, $\boldsymbol{u_n}$ is the nominal term, similar to a conventional model-based controller and $\boldsymbol{u_b}$ is the extension compensating for unpredictable uncertainties.

$$\boldsymbol{u} = \boldsymbol{u_n} + \boldsymbol{u_b} \tag{10}$$

with $\boldsymbol{u_n}, \boldsymbol{u_b}, \boldsymbol{u} \in R^n$.

Figure 23 illustrates the model-based control law by means of a block diagram. This figure depicts the addition of both controller-outputs, of the conventional controller and the BELBIC-

SMC-extension. Further, the constituents of the BELBIC-concept as well as their interlocking with the SMC-elements are detailed schematically.

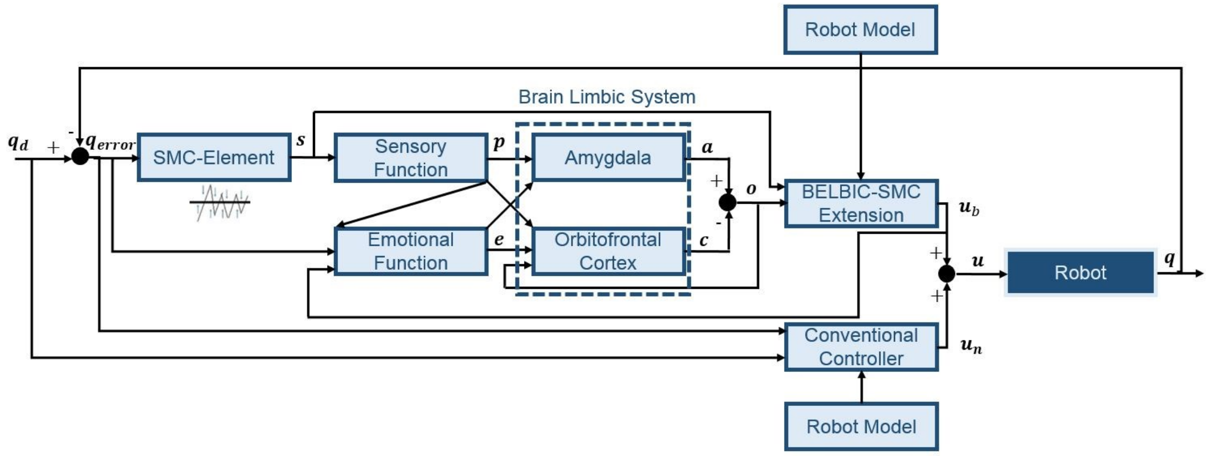



Figure 23: Block diagram of the BELBIC-SMC-control concept.

The equivalent control input is based on weighted position error-signals as in eq. (3) where $\boldsymbol{q_d}$ is the desired and $\boldsymbol{q}$ the actual position.

$$\ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_d - \boldsymbol{N}_1\dot{\boldsymbol{q}}_{error} \tag{11}$$

where $\boldsymbol{q}_{error}, \dot{\boldsymbol{q}}_{error}, \ddot{\boldsymbol{q}}_r \in R^n$ and $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in R^n$ express the desired or current link position, velocity and acceleration, respectively and $\boldsymbol{N_1} = n_1\boldsymbol{I_n}$ is an independent, constant square diagonal gain matrix with $n_1$ a constant factor dependent on the application and robot at stake.

In sliding mode control, a high-speed switching control law is used to first, guide the system's state trajectory onto a user-defined sliding surface in the state space and to, second keep it on this sliding surface for subsequent times. The sliding surface is defined in eq. (12) as a weighted sum of the position and the velocity errors:

$$\boldsymbol{s} = \dot{\boldsymbol{q}}_{error} + \boldsymbol{S}_1\boldsymbol{q}_{error} \tag{12}$$

where $\boldsymbol{S_1} = s_1\boldsymbol{I_n}$ is an independent, constant, positive-definite square diagonal gain matrix satisfying $s_1 < n_1$ and $s_1$ is a constant factor.

The equivalent term of the control input is expressed as follows:

$$\boldsymbol{u_n} = \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_r - \boldsymbol{N}_2\boldsymbol{q}_{error}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})(\dot{\boldsymbol{q}} - \boldsymbol{s}) + \boldsymbol{G}(\boldsymbol{q}) \tag{13}$$

where $\boldsymbol{N_2} = n_2\boldsymbol{I_n}$ is an independent, constant square diagonal gain matrix with $n_2$ a constant factor.

The second control input term $\boldsymbol{u_b}$ combines elements of a sliding mode and a BELBIC controller. Both concepts have been described in further details in 2.2. In this work robust sliding mode control is combined with learning BELBIC in order to combine the advantages of both concepts and cancel out their respective shortcomings. In the following the BELBIC-part is detailed. It is an action-generator (one output) based on sensory inputs and emotional cues (multiple inputs). If applicable, the learning algorithm starts from an arbitrary initialization.

The sensory input can be considered as a graded input-perception of the environment. $\boldsymbol{p} \in R^n$ is modelled as a weighted sliding surface term where $p_1$ is an arbitrary constant factor:

$$\boldsymbol{p} = p_1 \boldsymbol{s} \ . \tag{14}$$

Next to the generally neutral sensory input a reinforcing emotional cue function is implemented. The aspiration of this reward function being to maximize the reward, the maximum values of the suggested emotional cue function $\boldsymbol{e} \in R^n$ (eq. (15)) are situated in the targeted regions. This reinforcing input-term makes usage of error signals as well as weighted sensory input and control input terms:

$$\boldsymbol{e} = (\varepsilon_1 - \varepsilon_2)(\boldsymbol{p} - \epsilon \boldsymbol{u_b}) \tag{15}$$

where $\epsilon$ is a constant arbitrary factor. $\varepsilon_1, \varepsilon_2 \in R^1$ express the Cartesian distance between position- and velocity-errors of successive manipulator links. As an illustration: for 2 links:

$$\varepsilon_1 = \sqrt{{q_{error}}_1{}^2 + {q_{error}}_2{}^2} \ ,$$

$$\varepsilon_2 = \sqrt{\dot{q}_{error_1}{}^2 + \dot{q}_{error_2}{}^2} \ . \tag{16}$$

The amygdala with output $\boldsymbol{a} \in R^n$ can be considered as the actuator of the system as it maps sensory stimuli to their related emotional responses.

$$\boldsymbol{a} = \boldsymbol{p} \, g_a \tag{17}$$

where $g_a$ is a connection weight which is adjusted according to the gain function $\Delta g_a \in R^1$. This update law expresses proportionality to the difference between reward function and amygdala-output:

$$\Delta g_a = \gamma (\textstyle\sum_{j=1}^{n} (p_j \cdot \max(0, e_j - a_j))) \tag{18}$$

where $\gamma \in R^1$ is the associated constant learning rate. The sensory inputs to the amygdala are multiplied by the introduced weight function in a monotonic manner, i.e. the weights cannot decrease. This is inspired by the natural learning behaviour where all learned emotional responses are permanent and where the inhibition of no longer appropriate connections is the orbitofrontal cortex' duty.

The orbitofrontal cortex, the inhibitor reacts to environmental changes by inhibiting connections which were established by the amygdala but which are no longer relevant.

$$\boldsymbol{c} = \boldsymbol{p} \, g_c \tag{19}$$

$g_c$ is the connection weight adjusted following $\Delta g_c \in R^1$:

$$\Delta g_c = \beta (\boldsymbol{p} \cdot (\boldsymbol{o} - \boldsymbol{e})) \tag{20}$$

where $\beta \in R^1$ is the associated constant learning rate and $\beta > \gamma$.

The single output $\boldsymbol{o} \in R^n$ subtracts the inhibitory orbitofrontal outputs from the amygdaloidal outputs. It has the same sign as $\boldsymbol{s}$. It clarifies the interplay between actuator and inhibitor, between amygdala and orbitofrontal cortex:

$$\boldsymbol{o} = \boldsymbol{a} - \boldsymbol{c}\ . \tag{21}$$

In a next step the advantages of both control strategies are combined, resulting in a novel robust learning control scheme $\boldsymbol{u_b}$ (eq. (24)).

Time-dependent $b(t)$ (eq. (23)) is the main part of the intelligent controller-half $\boldsymbol{u_b}$ of the suggested control scheme $\boldsymbol{u}$. It combines the output of the BELBIC-controller $\boldsymbol{o}$ with the sliding surface $\boldsymbol{s}$ and a saturation function (eq. (22)) for chattering-suppression (Slotine, 1991) through element-by-element multiplication:

$$if\ |s_j| > \Delta_j: sats_j = sign(s_j) \quad else: sats_j = \frac{1}{\Delta_j} s_j \tag{22}$$

$$b = (\boldsymbol{sats} .* \ \boldsymbol{s}) \cdot \boldsymbol{o}\ \ , \tag{23}$$

$$\boldsymbol{u_b} = -\boldsymbol{M}(\boldsymbol{q})\boldsymbol{B_1} \int_0^T b(t)\, dt \tag{24}$$

with $j = 1, 2, \ldots, n$ indicating the link-number, $\Delta_1 = \Delta_2 = \cdots = \Delta_j = \Delta_n$ the boundary layer defining the region in which switch control is to be applied and $\boldsymbol{B_1}$ an independent, constant, positive-definite gain vector of length *n* and T the total time of the movement, $.*$ an element-by-element array multiplication.

Thus eq. (10) writes as eq. (25), by combining eq. (13) and eq. (24).

$$\boldsymbol{u} = \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{r}} - \boldsymbol{N_2}\boldsymbol{q}_{\boldsymbol{error}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})(\dot{\boldsymbol{q}} - \boldsymbol{s}) + \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{B_1} \int_0^T b(t)\, dt \tag{25}$$

To prove stable performance of the suggested control law, a Lyapunov candidate function $V$ is chosen (eq. (26)) and its time derivative (eq. (27)) is shown to be negative (Slotine, 1991).

$$V = \frac{1}{2}\boldsymbol{s^t}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{s} \qquad > 0 \tag{26}$$

Its time derivative

$$\dot{V} = \boldsymbol{s^t}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{s}} + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s} \tag{27}$$

can be rewritten using eq. (12) and (3) as eq. (28) and (29), respectively.

$$\dot{V} = \boldsymbol{s^t}\boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\boldsymbol{error}} + \boldsymbol{S_1}\dot{\boldsymbol{q}}_{\boldsymbol{error}}) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s} \tag{28}$$

$$\dot{V} = \boldsymbol{s^t}\boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_d + \boldsymbol{S_1}\dot{\boldsymbol{q}}_{\boldsymbol{error}}) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s} \tag{29}$$

Making use of eq. (8),

$$\dot{V} = \boldsymbol{s^t}(\boldsymbol{u} + \boldsymbol{Q}_i + \boldsymbol{d} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{M}(\boldsymbol{q})\boldsymbol{S_1}\dot{\boldsymbol{q}}_{\boldsymbol{error}}) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s} \tag{30}$$

of eq. (25),

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{M(q)}((\boldsymbol{\ddot{q}_r} - \boldsymbol{N_2 q_{error}}) + \boldsymbol{C(q,\dot{q})}(\boldsymbol{\dot{q}} - \boldsymbol{s}) + \boldsymbol{G(q)} - \boldsymbol{M(q)B_1}\int_0^T b(t)\,dt + \boldsymbol{Q}_i + \boldsymbol{d} - \boldsymbol{C(q,\dot{q})\dot{q}} - \boldsymbol{G(q)} - \boldsymbol{M(q)\ddot{q}_d} + \boldsymbol{M(q)S_1\dot{q}_{error}}\Big) + \frac{1}{2}\boldsymbol{s^t\dot{M}(q)s} \quad (31)$$

and of eq. (11), with $\boldsymbol{N_2} = (-\boldsymbol{S_1} + \boldsymbol{N_1})\boldsymbol{S_1}$, terms cancel out to

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{M(q)}(\boldsymbol{S_1} - \boldsymbol{N}_1)\boldsymbol{\dot{q}}_{error} + \boldsymbol{M(q)S_1}(\boldsymbol{S}_1 - \boldsymbol{N}_1)\boldsymbol{q}_{error} - \boldsymbol{C(q,\dot{q})s} - \boldsymbol{M(q)B_1}\int_0^T b(t)\,dt + \boldsymbol{Q}_i + \boldsymbol{d}\Big) + \frac{1}{2}\boldsymbol{s^t\dot{M}(q)s}\,. \quad (32)$$

With eq. (12)

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{M(q)}(\boldsymbol{S_1} - \boldsymbol{N}_1)\boldsymbol{s} - \boldsymbol{C(q,\dot{q})s} - \boldsymbol{M(q)B_1}\int_0^T b(t)\,dt + \boldsymbol{Q}_i + \boldsymbol{d}\Big) + \frac{1}{2}\boldsymbol{s^t\dot{M}(q)s}\,. \quad (33)$$

Using the skew-symmetry property P2

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{M(q)}(\boldsymbol{S_1} - \boldsymbol{N}_1)\boldsymbol{s} - \boldsymbol{M(q)B_1}\int_0^T b(t)\,dt + \boldsymbol{Q}_i + \boldsymbol{d}\Big) \quad (34)$$

and rearranging

$$\dot{V} = \boldsymbol{s^t M(q)}(\boldsymbol{S_1} - \boldsymbol{N}_1)\boldsymbol{s} - \boldsymbol{s^t M(q)B_1}\int_0^T b(t)dt + \boldsymbol{s^t Q}_i + \boldsymbol{s^t d}. \quad (35)$$

With the earlier introduced relation $s_1 < n_1$ (eq. (12)), $\boldsymbol{s^t M(q)(S_1 - N_1)s} \leq 0$.

For as far as term $-\boldsymbol{s^t M(q)B_1}\int_0^T b(t)\,dt$ is concerned: $\boldsymbol{sats}\,.*\,\boldsymbol{s}$ is shown to be always positive,

$$if\ |\boldsymbol{s}| > \Delta\text{:}\quad \boldsymbol{sats}\,.*\,\boldsymbol{s} = sign(\boldsymbol{s}) \cdot \boldsymbol{s} \quad > 0\,;\ else\text{:}\quad \boldsymbol{sats}\,.*\,\boldsymbol{s} = \boldsymbol{s^t}\frac{1}{\Delta}\boldsymbol{s} \quad > 0 \quad (36)$$

where Δ is the boundary layer defining the region in which switching control is to be applied.

$sign(\boldsymbol{o}) \neq sign(\boldsymbol{s})$ by definition. Integrand and integral are of the same sign. Then with $\boldsymbol{B_1}$ being positive-definite and property P1, the term $-\boldsymbol{s^t M(q)B_1}\int_0^T b(t)\,dt$ is always negative.

$\boldsymbol{s^t Q}_i \leq 0$ and $\boldsymbol{s^t d} \leq 0$ through the definition (5.1).

### 5.2.2 Results – Simulation

As a verification of the performance of the suggested control scheme, the controller is evaluated through simulation in a Matlab/Simulink-environment. For the simulation a two-link planar robotic manipulator with revolute joints as described in 4.1 is considered. Excellent simulation-results were obtained (Klecker, 2016c) (Klecker, 2017).

Robot-external disturbances are added as time-dependent 2-dimensional function. Internal uncertainties are added in the form of maximal +/-10% deviations from the dynamic parameters-values. Switching constraints are implemented in the form of a desired trajectory switching between different curved and straight segments as shown in Figure 24. The desired continuous trajectory is not differentiable.

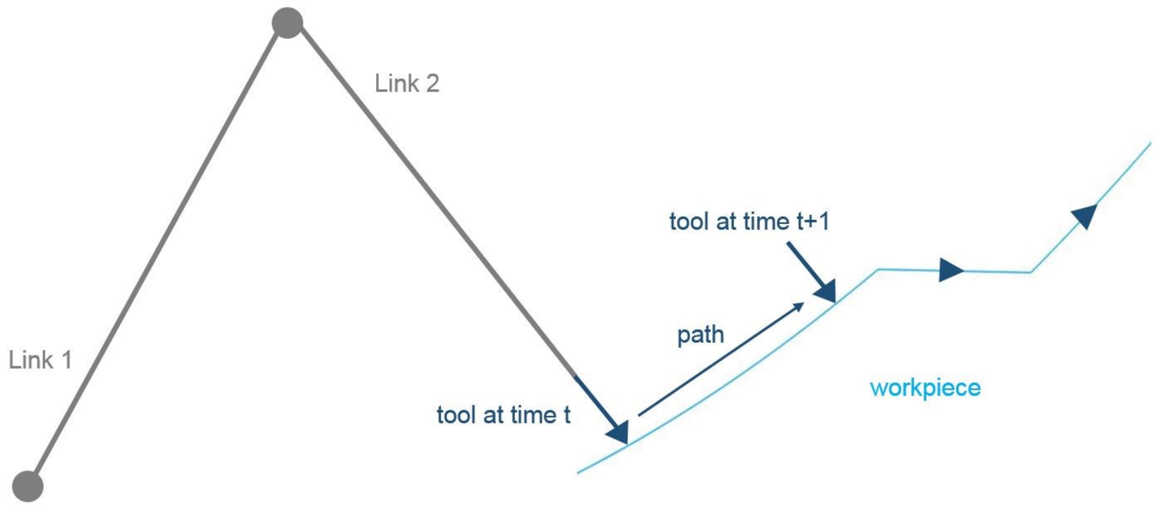
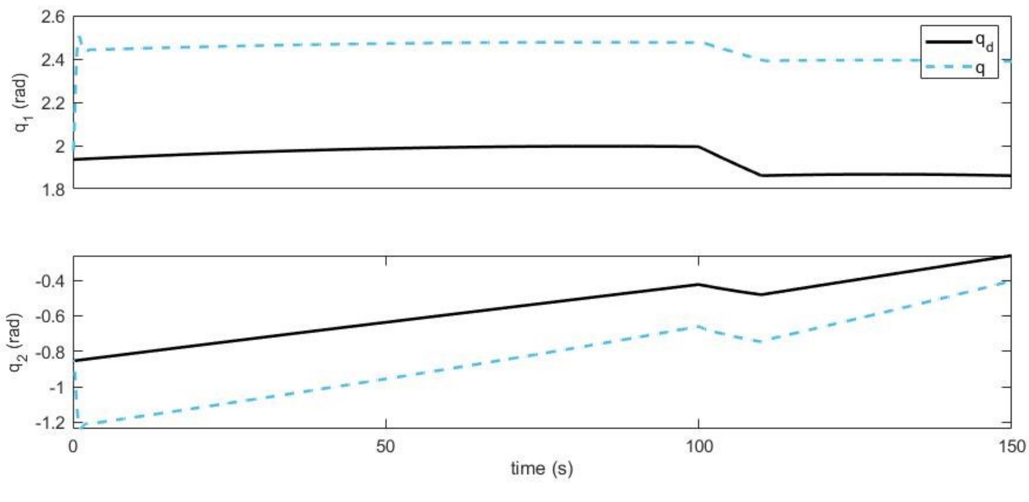



Figure 24: Trajectory to be tracked by the tool on the workpiece surface in the simulated application; the arrows indicate the movement-direction.

The numerical values for the parameters of the PID, CJT and BELBIC-SMC defined in 2.1.2 and 5.2.1: $d_p = 25, d_d = 10, d_i = 0, \Delta = 0.05,\ p_1 = 0.45,\ \gamma = 0.5,\ \epsilon = 0.5,\ \beta = 0.6,\ \boldsymbol{B_1} = \begin{bmatrix} 500 \\ 500 \end{bmatrix}$. $n_1, n_2, s_1$ are chosen freely as long as they obey the established characteristics. They are a function of the robot to be controlled as well as of the intended application. Here an iterative search-method starting from the Matlab-internal parameter-optimization as well as from manual tuning in ROS-simulations is applied (Klecker, 2017b) (Liu, 2011). For a straightforward comparison, for the PID and the CJT, the parameter-values of the BELBIC-SMC are transferred instead of individual gain-tuning.

Figure 25 and Table 2 show the performance of the three considered controllers for trajectory tracking of discontinuous freeform paths, exhibited on the example shown in Figure 24.



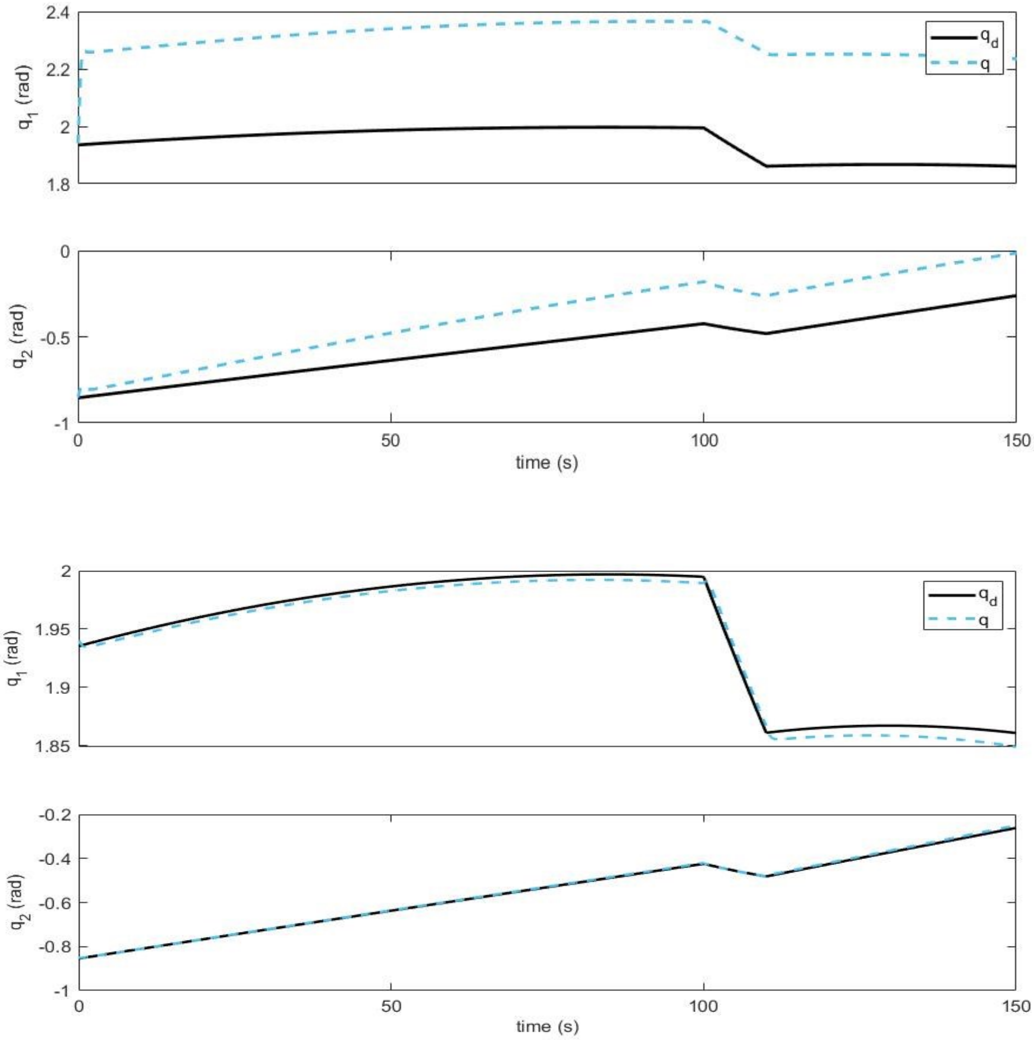



Figure 25: Position tracking of links 1 and 2 for the PID- (top), Computed Joint Torque- (middle) and BELBIC-SMC-controller (bottom).

Table 2: Absolute maximal, minimal and mean position errors for both manipulator-links using PID-, Computed Joint Torque (CJT)-control and BELBIC-SMC.

| | **PID** | | **CJT** | | **BELBIC- SMC** | |
|---|---|---|---|---|---|---|
| | $q_{error,1}$ | $q_{error,2}$ | $q_{error,1}$ | $q_{error,2}$ | $q_{error,1}$ | $q_{error,2}$ |
| **Max** [rad] | 0.5685 | 0.3877 | 0.3945 | 0.2477 | 0.0115 | 0.0122 |
| **Min** [rad] | 0.0046 | 0.0020 | 0.0046 | 0.0045 | 1.3567e-04 | 9.5147e-06 |
| **Mean** [rad] | 0.4938 | 0.2722 | 0.3513 | 0.1714 | 0.0053 | 0.0033 |

### 5.2.3 Results - Experimental Validation

In order to demonstrate the efficiency of the suggested controller in real-world applications beyond numerical simulation, an experimental validation is performed. The setup is described in 4.2. For the implementation of the model-based control concept, the values for the mass matrix and gravitational vector as provided in the FRI are used. Coriolis terms are neglected as no values are provided by the KUKA-internal program.

The first experiment is a goal-reaching task. The task is about moving the robot's joints consecutively to a specified goal position (Figure 26). Figure 27 shows the difference between the goal position and the currently measured angular position for the seven robot-joints over time. Table 3 shows that the BELBIC-SMC-concept outperforms PID- and Computed Joint Torque-control.

The second experiment is a path following application. With an input of a list of successive angular positions for each joint, the aim is to make the robot follow these target-positions consecutively. Figure 28 shows the start- and the end-positions of a zig-zag-movement. Table 4 presents the position tracking results for the BELBIC-SMC-concept.

Figure 26: End-effector movement.

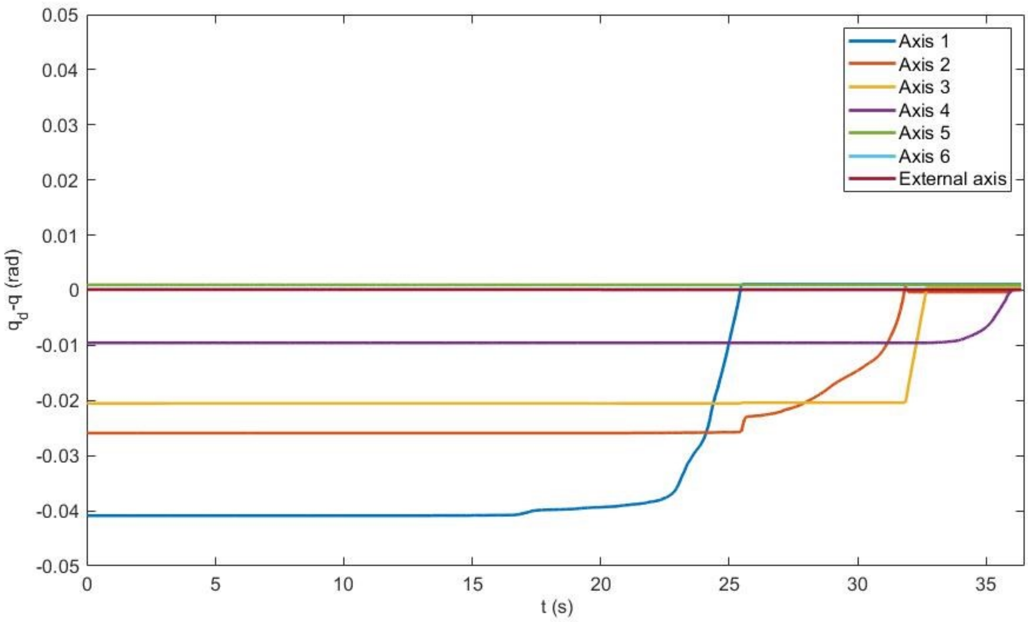



Figure 27: Position error for the 7 joints of the KUKA LWR4+ when reaching a desired goal position using the BELBIC-SMC-controller.

Table 3: Absolute maximal, minimal and mean position errors for all manipulator-links using PID-, Computed Joint Torque (CJT)-control and BELBIC-SMC when reaching a desired goal position.

| | | $q_{error,1}$ | $q_{error,2}$ | $q_{error,3}$ | $q_{error,4}$ | $q_{error,5}$ | $q_{error,6}$ | $q_{error,7}$ |
|---|---|---|---|---|---|---|---|---|
| **PID** | **Max** [rad] | 0.0409 | 0.0259 | 0.0205 | 0.0096 | 0.0010 | 0.0002 | 0.0002 |
| | **Min** [rad] | 0.0044 | 0.0070 | 0.0094 | 0.0095 | 0.0010 | 0.0002 | 0.0002 |
| | **Mean** [rad] | 0.0163 | 0.0206 | 0.0196 | 0.0096 | 0.0010 | 0.0002 | 0.0002 |
| | | $q_{error,1}$ | $q_{error,2}$ | $q_{error,3}$ | $q_{error,4}$ | $q_{error,5}$ | $q_{error,6}$ | $q_{error,7}$ |
| **CJT** | **Max** [rad] | 0.0409 | 0.0258 | 0.0250 | 0.0096 | 0.0019 | 0.0005 | 2.33451e-05 |
| | **Min** [rad] | 0.0003 | 0.0003 | 0.0087 | 0.0044 | 0.0010 | 0.0001 | 1.65384e-05 |
| | **Mean** [rad] | 0.0162 | 0.0182 | 0.0217 | 0.0075 | 0.0016 | 0.0003 | 2.19066e-05 |
| | | $q_{error,1}$ | $q_{error,2}$ | $q_{error,3}$ | $q_{error,4}$ | $q_{error,5}$ | $q_{error,6}$ | $q_{error,7}$ |
| **Belbic -SMC** | **Max** [rad] | 0.0409 | 0.0259 | 0.0205 | 0.0096 | 0.0010 | 0.0002 | 0.0001 |
| | **Min** [rad] | 0.0001 | 1.07288e-06 | 4.48376e-05 | 3.09944e-06 | 0.0010 | 0.0002 | 8.71101e-05 |
| | **Mean** [rad] | 0.0273 | 0.0212 | 0.0182 | 0.0092 | 0.0010 | 0.0002 | 9.35719e-05 |

Figure 28: End-effector movement.

Table 4: Absolute maximal, minimal and mean position errors for all manipulator-links using BELBIC-SMC when following a desired path.

| | $q_{error,1}$ | $q_{error,2}$ | $q_{error,3}$ | $q_{error,4}$ | $q_{error,5}$ | $q_{error,6}$ | $q_{error,7}$ |
|---|---|---|---|---|---|---|---|
| **Max** [rad] | 0.0067 | 0.1215 | 0.0057 | 0.1129 | 0.0004 | 0.0067 | 0.0091 |
| **Min** [rad] | 0.0010 | 2.07424e-05 | 5.3711e-05 | 0.0022 | 3.47793e-05 | 0.0003 | 4.77001e-05 |
| **Mean** [rad] | 0.0010 | 0.0147 | 0.0057 | 0.0024 | 0.0004 | 0.0004 | 4.94281e-05 |

### 5.2.4 Analysis and Summary

In 5.2, a solution to address the position control problem of trajectory tracking of discontinuous freeform-paths in the presence of uncertainties was developed.

The addressed path following problem can be situated in the area of automated manufacturing tasks. The control problem addressed occurs in all contact-based manufacturing processes, e.g. in welding, gluing and surface finishing. The considered path includes a succession of concave and straight trajectories. The need for the following of similar trajectories can be found for example in polishing or grinding of curved surfaces. Switching constraints and uncertainties are among the tackled intricacies. The former are due to tracking freeform parts with successions of convex, concave and straight surfaces. A switched nonlinear system is the result. Uncertainties of both internal and external nature are taken into account for the development of the control algorithms.

With conventional controllers this problem is not solvable. As model-free and model-based conventional controllers are only partially able to address complex trajectory tracking problems, an extension with a learning element is needed. A combination of BELBIC and SMC was developed. The latter was chosen because of its ability to robustly control nonlinear systems and because of its invariance with respect to disturbances. Sliding Mode Control however leads to chattering and requires unrealistic a priori knowledge of the tracking problem. To overcome these drawbacks, the bio-inspired BELBIC-based controller-extension was added.

In 5.2.1, the BELBIC-SMC-control algorithm was theoretically developed and its stability proven via a Lyapunov analysis.

As first validation of the suggested control concept, a numerical simulation was performed in the Matlab/Simulink-environment. The simulation verifies the tracking performance of the BELBIC-SMC-controller for the chosen case. There is a good alignment of the black solid line representing the desired position and the dotted blue line representing the tracking position (Figure 25). The BELBIC-SMC-controller outperforms both, the PID- and the Computed Joint Torque-controller. The freeform-shape and especially the discontinuity of the path deteriorate the tracking performance of the conventional controllers. Table 2 confirms the observations of Figure 25. The absolute mean position error is below the targeted 0.3 rad.

The second step in the validation process was the real-life implementation of the algorithm on a KUKA LWR 4+ robotic arm. Concerning the experiments (5.2.3) the error-signals converge to 0 (Figure 27). The position approaches the desired position for a goal-reaching-application when the BELBIC-SMC-controller is used. Table 3 shows that especially the minimal position error is significantly reduced for the BELBIC-SMC-controller. From Table 4 it is deduced that the average difference between desired and measured angular position never exceeds 0.015 rad and that the minimal error is practically zero for a path following-application. This is in compliance with the requirements established in 5.1.1.

The control algorithm presented in this part shows promising results, but its application is limited to well-known systems. The suggested concept assumes the robot dynamics are known a priori. Although the concept takes into account uncertainties in these robot model-parameters, it presumes partial knowledge of $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ as introduced in 5.1. In practice, these parameters are rarely known a priori. The lack of knowledge regarding these parameters would affect the efficiency of the control scheme and is addressed in the following sections.

## 5.3 BELBIC-SMC-NN

In 5.2, it has been confirmed that the combination of sliding mode control and BELBIC successfully addresses freeform trajectory tracking in the presence of switching constraints and uncertainties. Advantages of both concepts are fused while their disadvantages alleviate each other. The application of the suggested model-based controller however is limited to systems with a priori established robot parameters. $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ are rarely known and therefore have to be estimated. In this section, the control concept is extended for the case of unknown robot kinematics and dynamics. From the literature survey in 2.3, the hypothesis that a Radial Basis Function Neural Network is optimally chosen as estimator for unknown robot parameters is deduced.

### 5.3.1 Algorithm and Lyapunov-Stability Analysis

The concept suggested in 5.2 is extended for unknown robot kinematics and dynamics with a Gaussian Radial Basis Function (RBF) artificial Neural Network (NN). The latter is used to estimate the unknown robot model. The combination of the different control elements is illustrated in Figure 29.

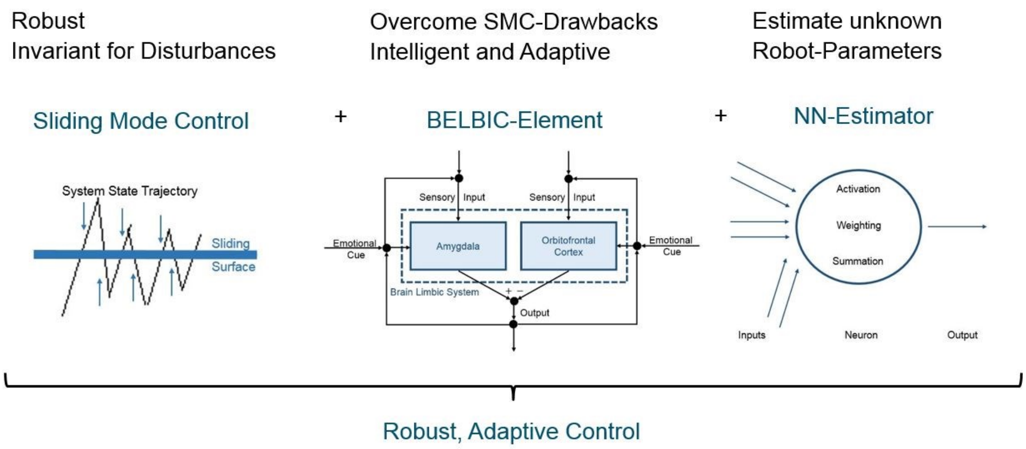



Figure 29: Combination of SMC-, BELBIC- and NN-elements for robust and adaptive control.

For the control of the nonlinear switched system described by eq. (8), as a starting point, the following control law is suggested:

$$\boldsymbol{u} = \boldsymbol{u_S} \tag{37}$$

where $\boldsymbol{u_S}$ is a conventional sliding mode control term with $\boldsymbol{u_S}, \boldsymbol{u} \in R^n$.

$$\boldsymbol{s} = \dot{\boldsymbol{q}}_{\boldsymbol{error}} + \boldsymbol{F}\, \boldsymbol{q}_{\boldsymbol{error}} \tag{38}$$

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}}_{\boldsymbol{error}} + \boldsymbol{F}\dot{\boldsymbol{q}}_{\boldsymbol{error}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{F}\dot{\boldsymbol{q}}_{\boldsymbol{error}} \tag{39}$$

The sliding surface $\boldsymbol{s} \in R^n$ and its time derivative are defined in eq. (38) and (39) as a weighted sum of the position and the velocity errors as in eq. (12) with $\boldsymbol{F} \in R^{nxn}$ a constant, positive-definite gain matrix. The saturation function $\boldsymbol{sat}$ of the sliding surface $\boldsymbol{s}$ is defined as in eq. (22). In the case of known robot dynamic parameters, i.e. $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ from eq. (8) are well identified, a conventional sliding mode control term can be expressed by

$$\boldsymbol{u_s} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{S}\boldsymbol{s} \tag{40}$$

where $\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{F}\, \boldsymbol{q}_{\boldsymbol{error}}$ and $\boldsymbol{S} \in R^{nxn}$ a constant, positive-definite gain matrix.

Further

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{s}} &= \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{F}\dot{\boldsymbol{q}}_{\boldsymbol{error}}) = \\ &-\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{r}} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{s} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{r}} - \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{d} + \boldsymbol{u} + \boldsymbol{Q}_i \quad . \end{aligned} \tag{41}$$

The non-linear robot function as defined in eq. (9) and (42) can be derived from eq. (40)

$$\boldsymbol{f} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G}(\boldsymbol{q}) \ . \tag{42}$$

Putting eq. (42) into eq. (41) and (40) leads to

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{s}} = -\boldsymbol{f} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{s} + \boldsymbol{d} + \boldsymbol{u} + \boldsymbol{Q}_i \ , \tag{43}$$

$$\boldsymbol{u_S} = \boldsymbol{f} + \boldsymbol{S}\boldsymbol{s} \ . \tag{44}$$

$\boldsymbol{f}$ as defined in eq. (42) represents the dynamics of the robotic manipulator to be controlled. The approach chosen here is to make use of $\hat{\boldsymbol{f}}$, the output of an RBF-NN designed to approximate $\boldsymbol{f}$.

Regarding the implemented 3-layer RBF-NN, the nonlinear activation function for node $j$ is illustrated by the following equation

$$h_j(x) = e^{-\|x - c_{ij}\|^2 / {b_j}^2} \tag{45}$$

where $\boldsymbol{h} = [h_1, h_2, \ldots, h_n]^T$ is the output of the Gaussian function, $i$ is the network´s input number, $j$ its number of hidden layer nodes, the norm is defined as the Euclidean distance, $\boldsymbol{c} = [c_{ij}]$ is the coordinate value of neural network $j$'s Gaussian function's centre point for input $i$, $b_j$ is its standard deviation, i.e. its width and $\boldsymbol{x} = [\boldsymbol{q_{error}}^T\ \dot{\boldsymbol{q}}_{\boldsymbol{error}}^{\boldsymbol{T}}\ \boldsymbol{q_d}^T\ \dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{T}}\ \ddot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{T}}]$ is the input of the neural network, chosen here with respect to $\boldsymbol{f}$.

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{h}(\boldsymbol{x}) \tag{46}$$

$$\dot{\boldsymbol{W}} = \boldsymbol{w}\boldsymbol{h}(\boldsymbol{x})\boldsymbol{s}^T \tag{47}$$

$$\hat{\boldsymbol{f}} = a\boldsymbol{y} \tag{48}$$

Eq. (46) defines the network's output with $\boldsymbol{W}$ a weight matrix. Eq. (47) presents the adaptive rule of the suggested RBF-NN with $\boldsymbol{w} \in R^{nxn}$being a positive constant matrix, $\boldsymbol{h}(\boldsymbol{x}) \in R^n$, the output vector of the hidden layer and $\boldsymbol{s}$ the sliding surface as defined in eq. (38). Eq. (48) estimates $\boldsymbol{f}$ with $a$ a positive constant.

Eq. (44) then becomes

$$\boldsymbol{u_S} = \hat{\boldsymbol{f}} + \boldsymbol{S}\boldsymbol{s} \ . \tag{49}$$

For eq. (43) this signifies

$$M(q)\dot{s} = -f - C(q,\dot{q})s + d + Q_i + \hat{f} + Ss$$
$$= -(C(q,\dot{q}) - S)s - \rho \quad (50)$$

with $\tilde{f} = f - \hat{f}$, $\rho = \tilde{f} - d - Q_i = a\widetilde{W}h(x) + e - d - Q_i$ and $\widetilde{W} = W - \widehat{W}$ where $\widehat{W}$ is the approximated weight matrix and $e$ is the approximation error of the neural network.

A Lyapunov analysis is used to identify stability issues of the conventional control system (Slotine, 1991). A Lyapunov candidate function $V$ whose time derivative has to be negative in order to guarantee stability is chosen in eq. (51).

$$V = \frac{1}{2}s^T M(q)s \quad (51)$$

$$\dot{V} = s^T M(q)\dot{s} + \frac{1}{2}s^T \dot{M}(q)s \quad (52)$$

Using eq. (50)

$$\dot{V} = -s^T C(q,\dot{q})s + s^T Ss - s^T \rho + \frac{1}{2}s^T \dot{M}(q)s \quad (53)$$

and property P2

$$\dot{V} = s^T Ss - s^T \rho. \quad (54)$$

As $s^T Ss \geq 0$, in order to guarantee $\dot{V} \leq 0$, $s^T \rho \geq s^T Ss$ has to be true.

From eq. (54) it can be deduced that the system-stability depends on the term $s^T \rho$, mainly on $\rho$, i.e. on the approximation-accuracy and the impact of both external disturbances and the switched nonlinear system.

In order to compensate for this destabilizing term, a robustifying control extension $u_R \in R^n$ (eq. (55)) is added. The practical confirmation of this hypothesis results from the validation through simulation in 5.3.2.

$$u_R = \hat{f}.* (R_1 s.* sats + R_2 \int_0^T p.* o\, dt) \quad (55)$$

with $R_1, R_2 \in R^{nxn}$ constant, positive-definite gain matrices. $\int_0^T p.* o\, dt$ of eq. (55) was defined in the BELBIC-paragraph in 5.2. The goal of $u_R$ is to compensate for uncertainties and remediate the destabilizing effects of the NN-approximation error $e$.

Finally, the conventional sliding mode control term $u_S$ and the robustifying bio-inspired control term $u_R$ are merged in the following control law

$$u = u_S + u_R. \quad (56)$$

Figure 30 illustrates the suggested control law by means of a block diagram. This figure depicts the addition of both controller-outputs, of the SMC-controller and the robustifying extension. For the latter, the bio-inspired constituents as well as their interlocking with the SMC-elements are detailed schematically. Further, the extension with the radial basis function artificial neural network estimating the robot model for both controller-parts is illustrated.

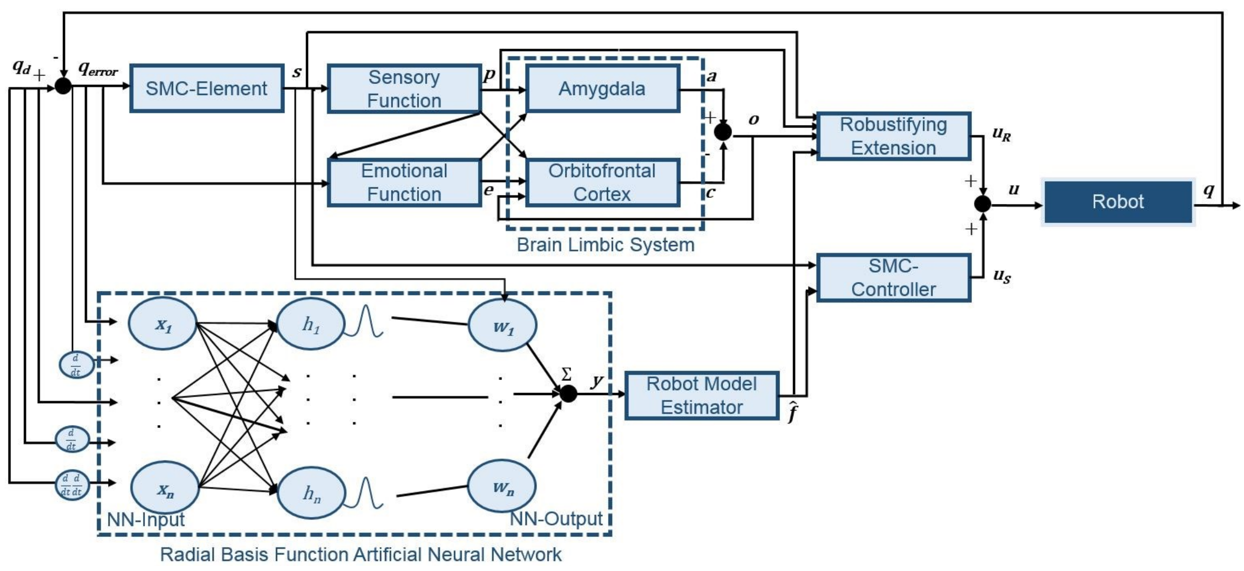



Figure 30: Block diagram of the BELBIC-SMC-NN-control concept.

### 5.3.2 Results – Simulation

As a verification of the performance of the suggested control scheme, the controller is verified through simulation in a Matlab/Simulink-environment. Good results were obtained in (Klecker, 2017c). For the simulation a two-link planar robotic manipulator with revolute joints as described in 4.1 is considered. In order to allow a comparison with the performance-results of the conventional PID-, computed joint torque CJT-controllers as well as the BELBIC-SMC-concept described in 5.2, the same trajectory tracking application as chosen and introduced in 5.2 is used here.

The parameters of the RBF Neural Network as defined in 5.3.1:

$$\boldsymbol{c} = \begin{matrix} -0.21 & -0.14 & -0.07 & 0 & 0.07 & 0.14 & 0.21 \\ -0.21 & -0.14 & -0.07 & 0 & 0.07 & 0.14 & 0.21 \\ -0.21 & -0.14 & -0.07 & 0 & 0.07 & 0.14 & 0.21 \\ -0.21 & -0.14 & -0.07 & 0 & 0.07 & 0.14 & 0.21 \\ -0.21 & -0.14 & -0.07 & 0 & 0.07 & 0.14 & 0.21 \end{matrix},$$

$b = 1.2589$, the constant positive factor in matrix $w$ is $15$, $a = 0.3$, number of nodes for the hidden layer = 7. These parameters are determined using an iterative search-method suggested in (Liu, 2011) (Liu, 2013) and starting from the parameters determined in 5.2.2. The parameters of the control algorithm: $\boldsymbol{F} = 3.58\boldsymbol{I_2}$, $\Delta = 0.05$, $\boldsymbol{S} = 20\boldsymbol{I_2}$, $\mathrm{R}_1 = 6.5\mathrm{I}_2$, $\mathrm{R}_2 = 50\mathrm{I}_2$, $\boldsymbol{N_1} = 10\boldsymbol{I_2}$, $\boldsymbol{N_2} = 35\boldsymbol{I_2}$, $\boldsymbol{N_3} = 7\boldsymbol{I_2}$, $p_1 = 0.45$, $\epsilon = 0.5$, $\gamma = 0.5$, $\beta = 0.6 > \gamma$.

Figures 31 and 32 exhibit the position tracking performance of the suggested control concept for the following of discontinuous freeform paths, exhibited on the example shown in Figure 24.

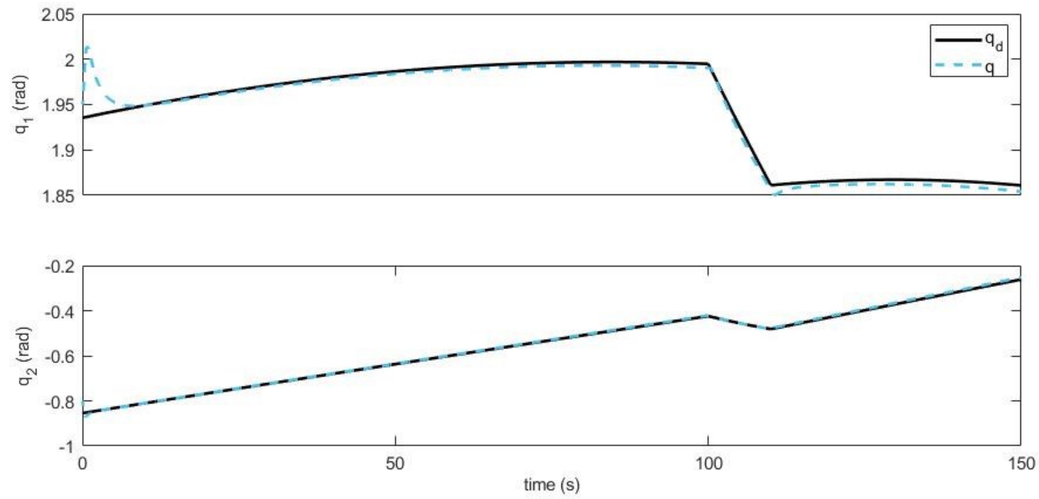



Figure 31: Position tracking of link 1 (top) and link 2 (bottom).

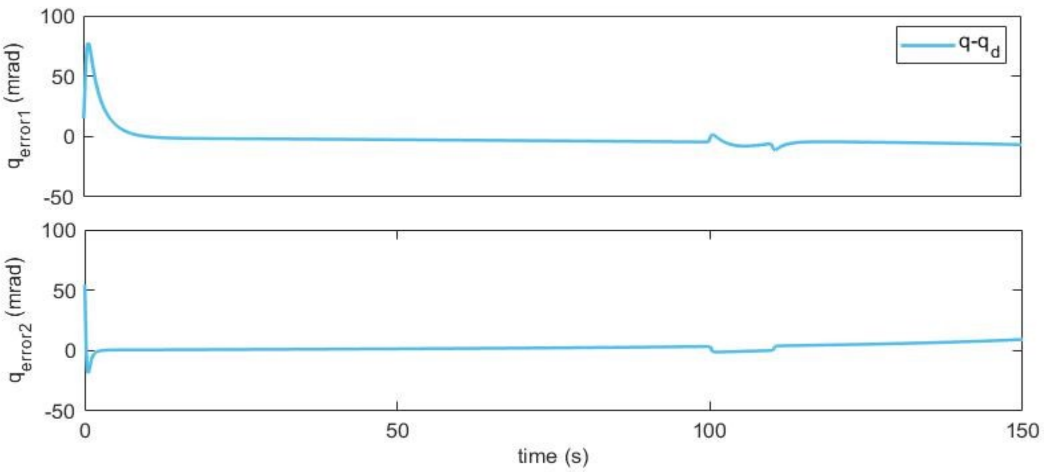



Figure 32: Position error of link 1 (top) and link 2 (bottom).

### 5.3.3 Analysis and Summary

In 5.3, the combination of SMC and BELBIC suggested in 5.2 to address the position control problem of trajectory tracking of discontinuous freeform-paths in the presence of uncertainties, was extended for the case of unknown robot parameters.

The parameters $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ as introduced in eq. (8) are rarely known. Model-based controllers, e.g. BELBIC-SMC suggested in 5.2 however make use of these matrices, vectors and require them for a successful performance. Therefore they need to be estimated, i.e. an estimator for robot model parameters has to be included in the BELBIC-SMC algorithm.

In this control concept, instead of using the parameters $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$, they are grouped into a single robot-function $\boldsymbol{f}$ (eq. (9) or (42)). The unknown robot model is approximated as a nonlinear robot-function $\boldsymbol{f}$. Compared to estimating the parameters $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ separately, the computational cost and the resulting approximation error can be reduced when the parameters are estimated jointly as one function.

Based on the literature survey in 2.3, it is hypothesized that a simple Gaussian Radial Basis Function Neural Network is sufficient for estimating the unknown robot function. The artificial neural network which was implemented in this control scheme is a plain perceptron-model which is sufficient for the considered application. Its advantages include: it is fast and no training phase is required during operation. If necessary, for future applications a more advanced, performant neural network can replace the current artificial neural network. In this case, it should be taken into account that training time and data need to be provided.

In 5.3.1, the algorithm was theoretically developed.

For validation purposes, a numerical simulation was performed in the Matlab/Simulink-environment. For as far as the simulation is concerned (5.3.2) it verifies the tracking performance of the controller for the chosen case despite the lack of knowledge of the robot parameters. There is a good correspondence between the black solid line representing the desired position and the dotted blue line representing the tracking position. The controller outperforms the model-free PID-controller (2.1.2) as was shown in Table 2. A major improvement can be observed in the trajectory tracking from the conventional PID-controller to the bio-inspired concept. Compliant to the requirements set in 5.1.1, though improvements are seen for maximal and minimal trajectory tracking errors, the most significant improvement is observed for the mean performance.

The suggested BELBIC-SMC-NN-control scheme involves a non-negligible number of parameters which have to be either known a priori or determined through trial-and-error. This is admissible for a relatively simple use case and robot as analysed in the simulation in 5.3.2. For more complex systems as considered in the real-life experimental validation, however, this process is too time consuming and not straightforward enough. Despite promising theoretical and simulation-results, the application of the BELBIC-SMC-NN-control is limited to rather simple systems as the suggested Neural Network requires a priori determination of its parameters. To allow the practical implementation of the suggested control-concept in a wider range of applications, the following section focuses on reducing the dependence on a priori determined parameters for the algorithm.

## 5.4 Combination of SMC and Learning with Adaptive NN

The application of the suggested controller (5.3) is limited to comparatively simple systems as it involves a non-negligible number of fixed parameters. Besides not being adaptive to changes in the environment, these parameters have to be either known a priori or determined through highly time-consuming trial-and-error. In this section, the focus lies on relieving the dependence on fixed a priori determined and initialized parameters. More precisely, it is hypothesized that an interconnected combination of SMC, reinforcement learning-based learning element and RBF-NN with adaptive parameters complies with the requirements.

### 5.4.1 Algorithm and Lyapunov-Stability Analysis

The trajectory tracking control problem for the case of unknown robot models is adapted in order to facilitate its practical implementation in real-life scenarios. By adapting the artificial neural network and the interconnection of the involved control-concepts, the number of parameters which need to be known a priori is reduced and consequently the range of application of the suggested control scheme is expanded (Klecker, 2018b).

The suggested control concept (eq. (57)) is composed of a conventional controller $\boldsymbol{u}_c \in R^n$ and a bio-inspired learning controller-extension $\boldsymbol{u}_e \in R^n$ to enhance the system-performance as well as its robustness.

$$\boldsymbol{u} = \boldsymbol{u}_c + \boldsymbol{u}_e \tag{57}$$

Sliding mode control is at the base of the suggested controller. The sliding surface $\boldsymbol{s} \in R^n$ is defined in eq. (58).

$$\boldsymbol{s} = \dot{\boldsymbol{q}}_{\boldsymbol{error}} + \boldsymbol{q}_{\boldsymbol{error}} \tag{58}$$

The conventional controller which is implemented for its robustness and invariance with respect to uncertainties is formalized in eq. (59).

$$\boldsymbol{u}_c = \boldsymbol{f} + \boldsymbol{c_c s} \tag{59}$$

where $\boldsymbol{f}$ was defined in eq. (9) and $\boldsymbol{c_c} \in R^{nxn}$ is a gain matrix.

The conventional controller is complemented by a bio-inspired controller-extension. This element is added because of its learning capacities and ability to address variations in the environment. It is inspired on reinforcement learning in general and the human learning process based on adaptive motivation-lifecycles in specific.

The key-elements of the concept are

- the incentive, I, based on the weighted perception of the agent's environment and serving as motivation for a learning system,
- the complementary action of an actuator, A, and a preventer, P, resulting in an adapted goal-directed output.

The structure of this concept is illustrated in Figure 33.

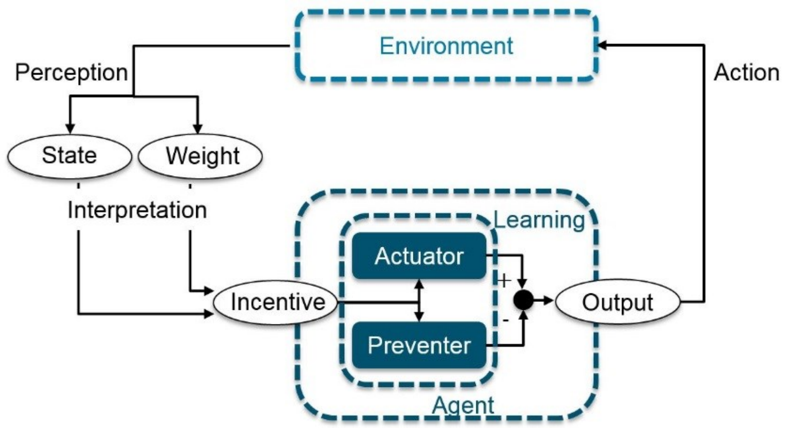



Figure 33: The principle of the learning concept.

In equation-form, the learning concept is expressed by eq. (60)-(66). The weighted perception of the agent's environment is defined as the position error (eq. (60)).

$$\boldsymbol{w_{state}} = \boldsymbol{q_{error}} = \boldsymbol{q} - \boldsymbol{q_d} \tag{60}$$

The incentive, I, is the motivation which serves as input to the learning system of the agent and follows eq. (61). $\boldsymbol{o} \in R^n$ (eq. (66)), the meaningful learning output is defined as the difference between the outputs of the actuator $\boldsymbol{a} \in R^n$ (eq. (62)) and the preventer $\boldsymbol{p} \in R^n$ (eq. (64)). The latter are updated according to the learning rates (eq. (63) and (65)). Learning reorganizes information as a combination of associating and predicting. The expectancy of a state and the understanding of an action-perception causation are at the base of the algorithm.

$$i_j = sgn(w_{state_j})\dot{w}_{state_j}(s_j - o_j) \tag{61}$$

$$\boldsymbol{a} = \boldsymbol{w_{state}}.* \Delta_{\boldsymbol{a}} \tag{62}$$

$$\Delta_{a_j} = c_a \dot{w}_{state_j} \max(0, i_j) \tag{63}$$

$$\boldsymbol{p} = \boldsymbol{w_{state}}.* \Delta_{\boldsymbol{p}} \tag{64}$$

$$\Delta_{\boldsymbol{p}} = c_p \dot{\boldsymbol{w}}_{\boldsymbol{state}}.*(\boldsymbol{o} - \boldsymbol{i}) \tag{65}$$

$$\boldsymbol{o} = \boldsymbol{a} - \boldsymbol{p} \tag{66}$$

where .* denotes element-wise multiplication, $j$ indicates the considered link and $c_a, c_p > 0$ are constant gain factors.

The output of the controller-extension $\boldsymbol{u_e}$ is expressed in eq. (67).

$$\boldsymbol{u_e} = \boldsymbol{f}.* \left(\boldsymbol{s}.* \boldsymbol{sats} + \int_0^T \boldsymbol{o}(t)\ dt\right) \tag{67}$$

The saturation function $\boldsymbol{sats} \in R^n$ was introduced in eq. (22) to increase the resistance to chattering of a sliding mode controller. T is the total time of the process. The output of the learning element is time-dependent in the sense that it changes as time passes and the robotic manipulator moves.

To prove stable performance of the suggested control law, a Lyapunov candidate function $V$ is chosen (eq. (68)) and its time derivative (eq. (69)) is shown to be negative (Slotine, 1991).

$$V = \frac{1}{2}\boldsymbol{s^t M(q) s} \qquad > 0 \tag{68}$$

Its time derivative

$$\dot{V} = \boldsymbol{s^t M(q)\dot{s}} + \frac{1}{2}\boldsymbol{s^t \dot{M}(q) s} \tag{69}$$

can be rewritten using eq. (58), (3) and (8), respectively

$$\dot{V} = \boldsymbol{s^t M(q)}(\boldsymbol{\ddot{q}_{error}} + \boldsymbol{\dot{q}_{error}}) + \frac{1}{2}\boldsymbol{s^t \dot{M}(q) s} \tag{70}$$

$$\dot{V} = \boldsymbol{s^t}(\boldsymbol{M(q)\ddot{q}} - \boldsymbol{M(q)\ddot{q}_d} + \boldsymbol{M(q)\dot{q}_{error}}) + \frac{1}{2}\boldsymbol{s^t \dot{M}(q) s} \tag{71}$$

$$\dot{V} = \boldsymbol{s^t}(\boldsymbol{u} + \boldsymbol{Q}_i + \boldsymbol{d} - \boldsymbol{C(q,\dot{q})\dot{q}} - \boldsymbol{G(q)} - \boldsymbol{M(q)\ddot{q}_d} + \boldsymbol{M(q)\dot{q}_{error}}) + \frac{1}{2}\boldsymbol{s^t \dot{M}(q) s}. \tag{72}$$

With eq. (57), (59) and (67)

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{f} + \boldsymbol{c_c s} + \boldsymbol{f}.*(\boldsymbol{s}.*\boldsymbol{sats} + \int_0^T \boldsymbol{o}(t)\ dt) + \boldsymbol{Q}_i + \boldsymbol{d} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}_{\boldsymbol{error}}\Big) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s}\,. \tag{73}$$

Using eq. (9) with $\boldsymbol{F} = \boldsymbol{I_n}$ in the previous equation

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}_{\boldsymbol{error}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{q}_{\boldsymbol{error}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{c_c s} + \boldsymbol{f}.*(\boldsymbol{s}.*\boldsymbol{sats} + \int_0^T \boldsymbol{o}(t)\ dt) + \boldsymbol{Q}_i + \boldsymbol{d} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}_{\boldsymbol{error}}\Big) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s} \tag{74}$$

some elements cancel out and further

$$\dot{V} = \boldsymbol{s^t}\Big(-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{s} + \boldsymbol{c_c s} + \boldsymbol{f}.*(\boldsymbol{s}.*\boldsymbol{sats} + \int_0^T \boldsymbol{o}(t)\ dt) + \boldsymbol{Q}_i + \boldsymbol{d}\Big) + \frac{1}{2}\boldsymbol{s^t}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{s}. \tag{75}$$

Making use of property P2 (5.1)

$$\dot{V} = \boldsymbol{s^t}\Big(\boldsymbol{c_c s} + \boldsymbol{f}.*(\boldsymbol{s}.*\boldsymbol{sats} + \int_0^T \boldsymbol{o}(t)\ dt) + \boldsymbol{Q}_i + \boldsymbol{d}\Big) \tag{76}$$

$\boldsymbol{s^t Q}_i \leq 0$, $\boldsymbol{s^t d} \leq 0$ through the definition introduced in 5.1 make up for $\boldsymbol{s^T c_c s} \geq \boldsymbol{0}$. The remaining terms affecting the sign of $\dot{V}$ are $\boldsymbol{s}^T\boldsymbol{f}.*\boldsymbol{sats}.*\boldsymbol{s}$ or rather $\boldsymbol{f}.*\boldsymbol{sats}$ and $\boldsymbol{s}^T\boldsymbol{f}.*\int_0^T \boldsymbol{o}(t)\ dt$. Because integral and integrand are of the same sign and $sign(\boldsymbol{o}) \neq sign(\boldsymbol{s})$ by definition, the stability depends on the robot dynamic parameters and in this case on their approximation.

As the non-linear robot function $\boldsymbol{f}$ is often unknown, it has to be estimated with a RBF-NN with j nodes in model-based controllers as described in 5.3. The non-linear Gaussian activation function for node j of network input i is defined in eq. (77).

$$h_j(\boldsymbol{x}) = e^{-\|\boldsymbol{x}-c_{ij}\|^2/{b_j}^2} \tag{77}$$

with $\boldsymbol{x} = [\boldsymbol{q}^T{}_{\boldsymbol{error}}, \dot{\boldsymbol{q}}^T{}_{\boldsymbol{error}}, \boldsymbol{q}^T{}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}^T{}_{\boldsymbol{d}}, \ddot{\boldsymbol{q}}^T{}_{\boldsymbol{d}}]$ the input of the network selected in the scope of $\boldsymbol{f}$, the to be approximated function. $c_{ij}$ is the coordinate value of node j's Gaussian function's centre point for input i and $b_j$ is the Gaussian function's width.

The aim is to facilitate the implementation of the controller. Therefore the number of fixed parameters and the system-dependence on them is reduced compared to the RBF-NN suggested in 5.3. To overcome the problems associated with a priori fixed centres and widths of the radial basis functions, they are updated online according to eq. (78) and (79).

$$\dot{c}_{ij} = c_{0_{:,j}}|s| \tag{78}$$

$$b_j = b_{0_j} + |1/s_j| \tag{79}$$

where $\boldsymbol{c_0} \in R^{jxi}$, $\boldsymbol{b_0} \in R^n$ are arbitrarily selected initializations and $s_j \neq 0$.

The approximation of $\boldsymbol{f}$, $\hat{\boldsymbol{f}}$ is computed as output of the RBF-NN (eq. (80)).

$$\hat{\boldsymbol{f}} = \boldsymbol{Wh}(\boldsymbol{x}) \tag{80}$$

where $\boldsymbol{W}$ is the weight which is adapted according to the update law (eq. (81)).

$$\dot{\boldsymbol{W}} = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{s}^T \tag{81}$$

The interconnection of the three elements, i.e. SMC, bio-inspired learning and adaptive RBF-NN combined in the suggested concept is graphically represented in Figure 34.

The combination and interplay of the elements previously described in equation-form are further illustrated in Figure 35. The structure of the suggested controller is depicted and clarified as a block diagram.

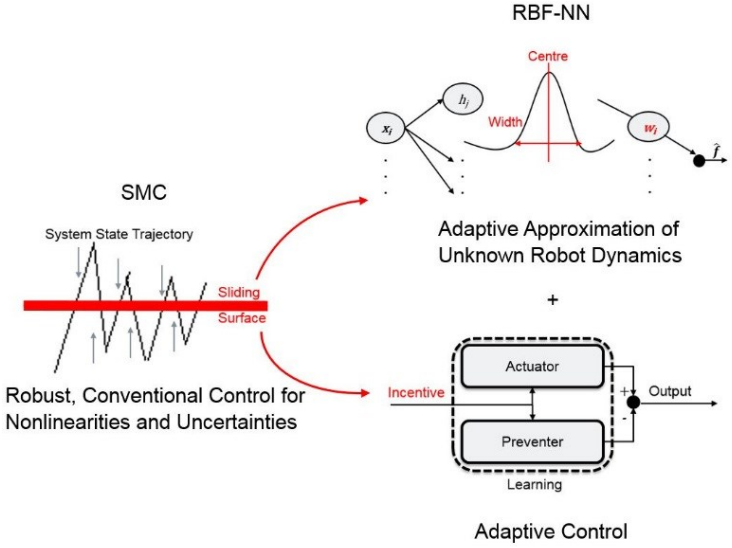



Figure 34: Interconnection of SMC, RBF-NN and adaptive learning.

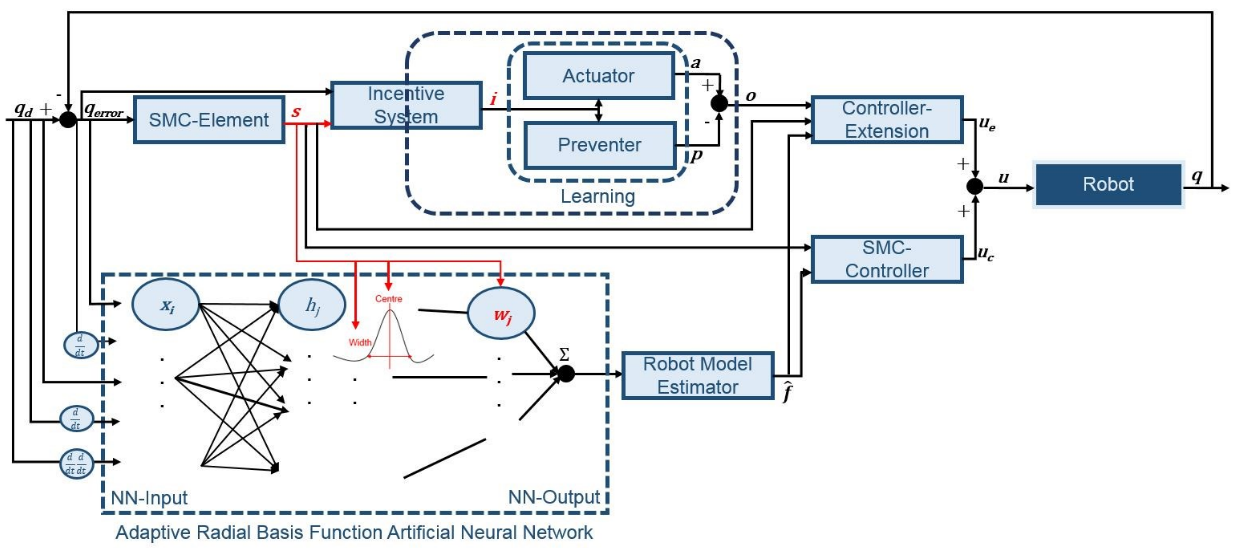



Figure 35: Block diagram of the control concept.

### 5.4.2 Results - Simulation

As a first validation of the suggested control concept, a basic path following case is simulated on the planar robot described in 4.1. The inputs for the considered path following application are two .csv-files, i.e. lists of successive desired joint positions in radians. The parameter-values were consciously and arbitrarily selected small and simple to demonstrate the controller-performance independently of specific parameter-values. $\boldsymbol{c_c} = 10\boldsymbol{I_2}$, $c_a = 5$, $c_p = 5$ and for the initialization of $\boldsymbol{c}, \boldsymbol{b}$: $\boldsymbol{c_{0_{:,j}}} = [-0.999, -0.666, -0.333, 0, 0.333, 0.666, 0.999]$ and $\boldsymbol{b_0} = [1, 1]$.

The simulation results for the path following application are depicted in Figures 36 and 37. They show the position tracking error for both robot links.

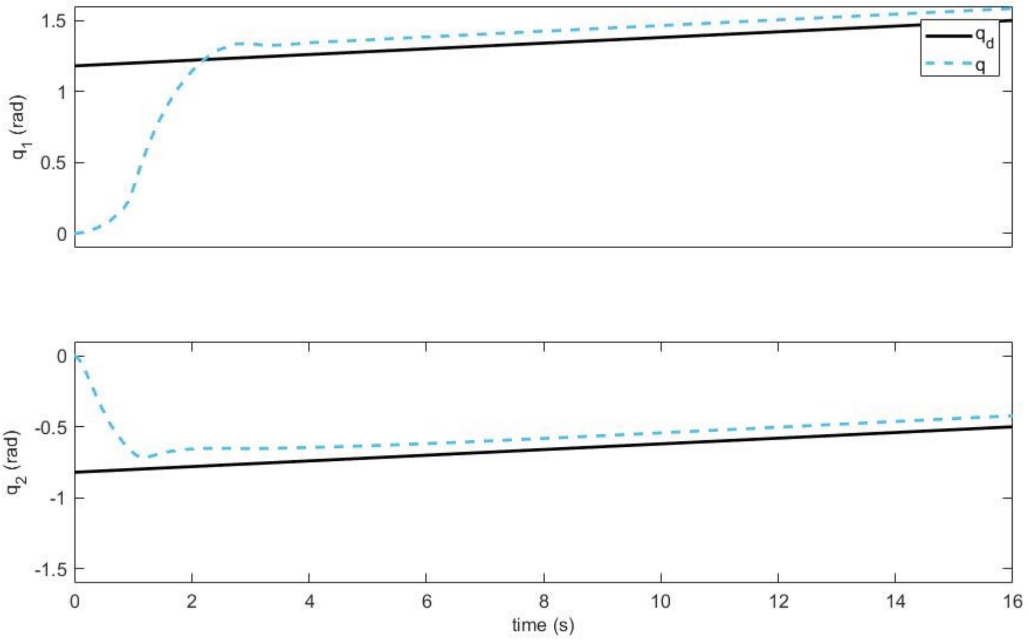



Figure 36: Position tracking of link 1 (top) and link 2 (bottom).

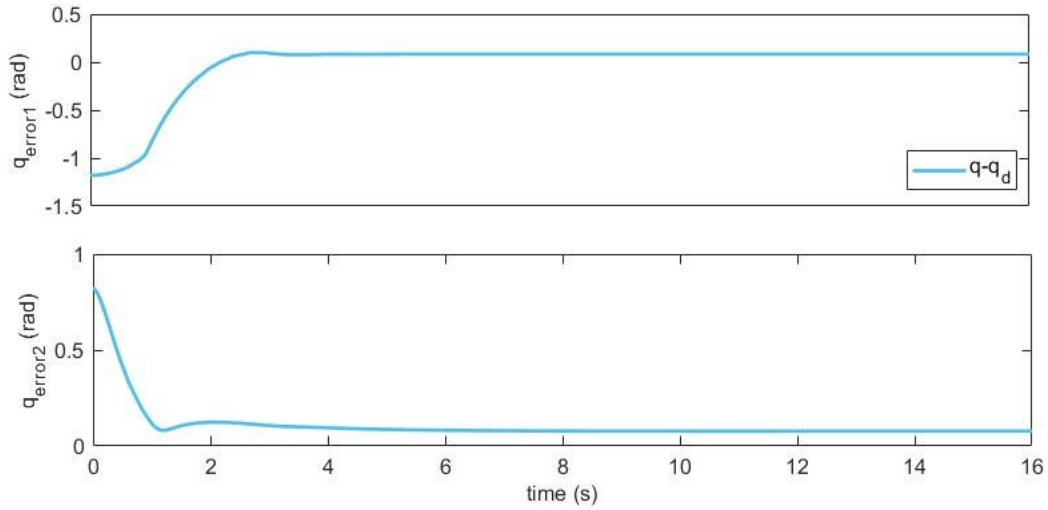



Figure 37: Position error of link 1 (top) and link 2 (bottom).

### 5.4.3 Results – Experimental Validation

The main validation of the suggested control concept is done experimentally. The experiments are performed on the 7 DOF-KUKA LWR 4+-robot described in 4.2.

The first experiment is a uniaxial trajectory tracking use case: the end-effector has to follow a straight line on a plane surface. Figure 38 illustrates the movement of the tool and Figure 39 shows the position error of the moved joint. The latter converges to 0 and the joint attains the target.

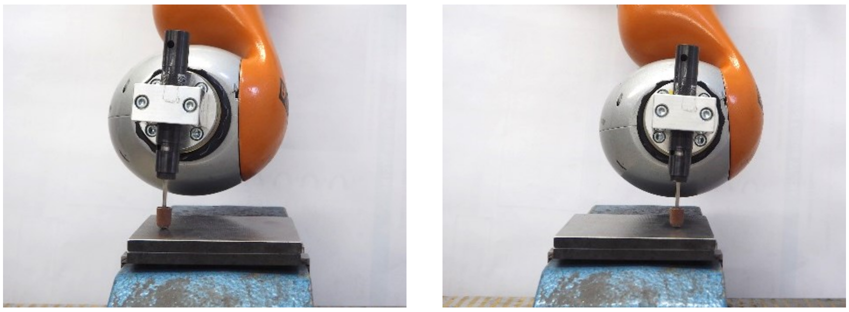


Figure 38: End-effector movement.

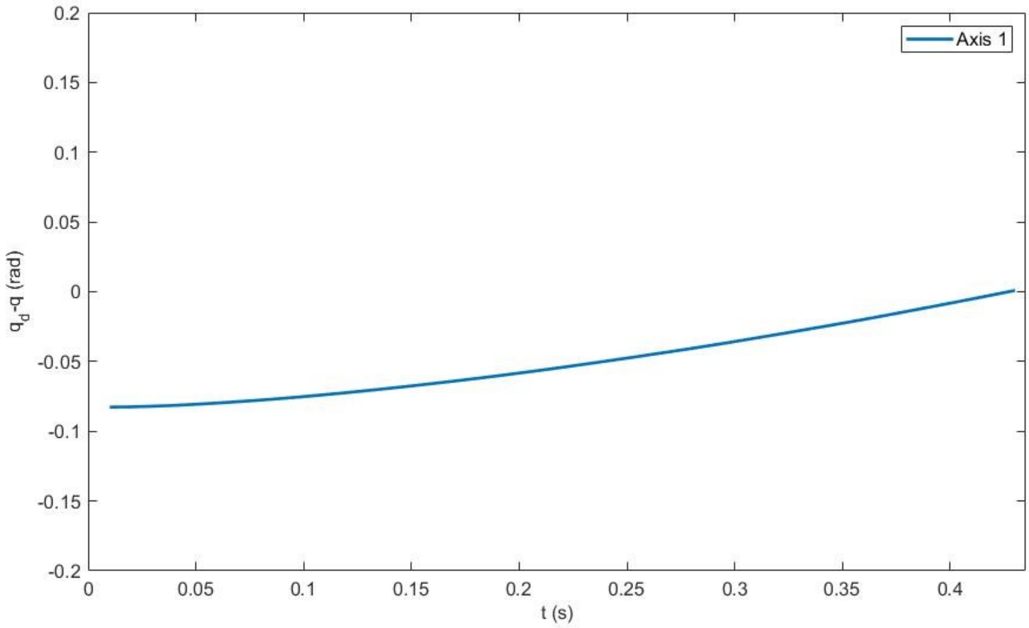



Figure 39: Position error.

The second use case involves moving from a contact situation to free space. Figure 40 shows the end effector-movement: the tool is in contact with the workpiece-surface (left), is lifted off and reaches its goal position in free space (right). Figure 41 represents the joint angular position errors of all 7 robot joints which all converge to 0.

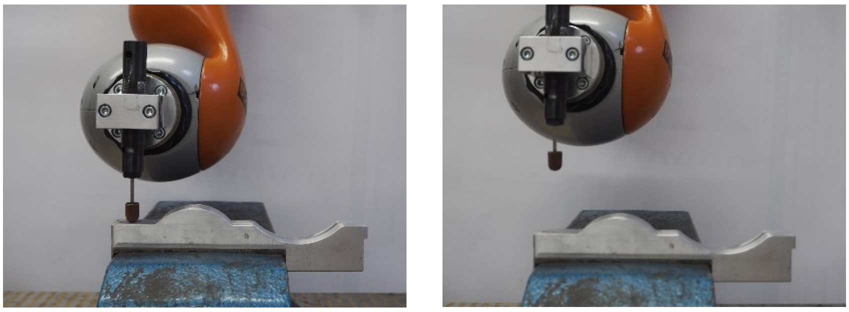


Figure 40: End-effector movement.

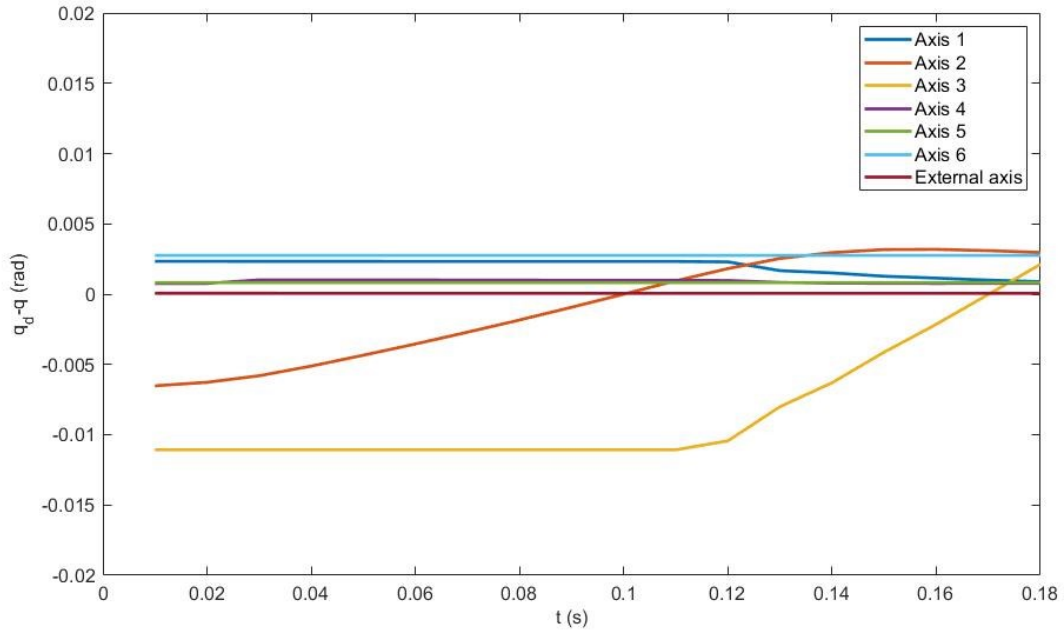



Figure 41: Position error.

The final use case consists in reaching a position in free space. The inputs for the goal-reaching application are desired positions for all 7 joints. Figure 43 shows the position error for the joints over time. All curves converge to 0 which means that the robot has reached the desired position.

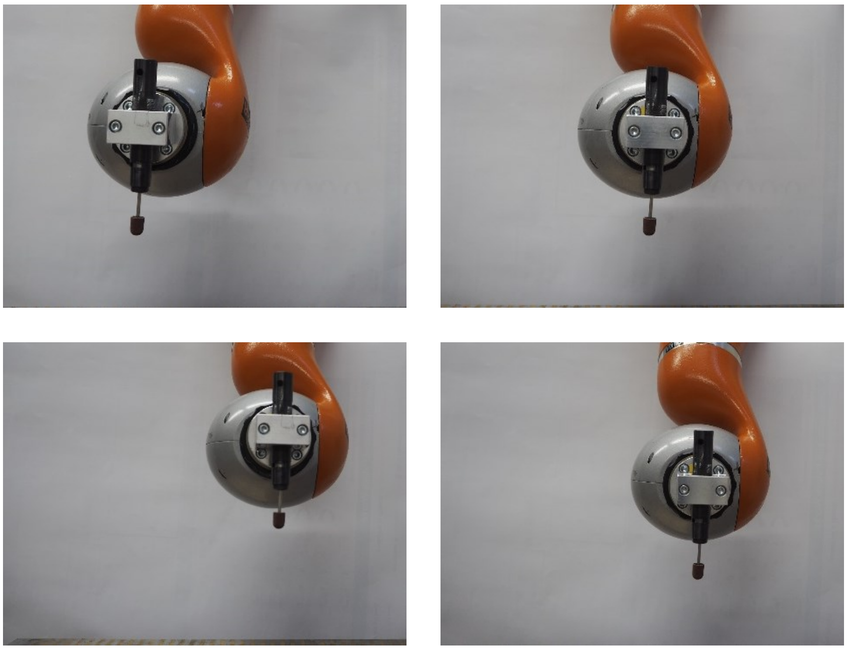


Figure 42: End-effector movement.

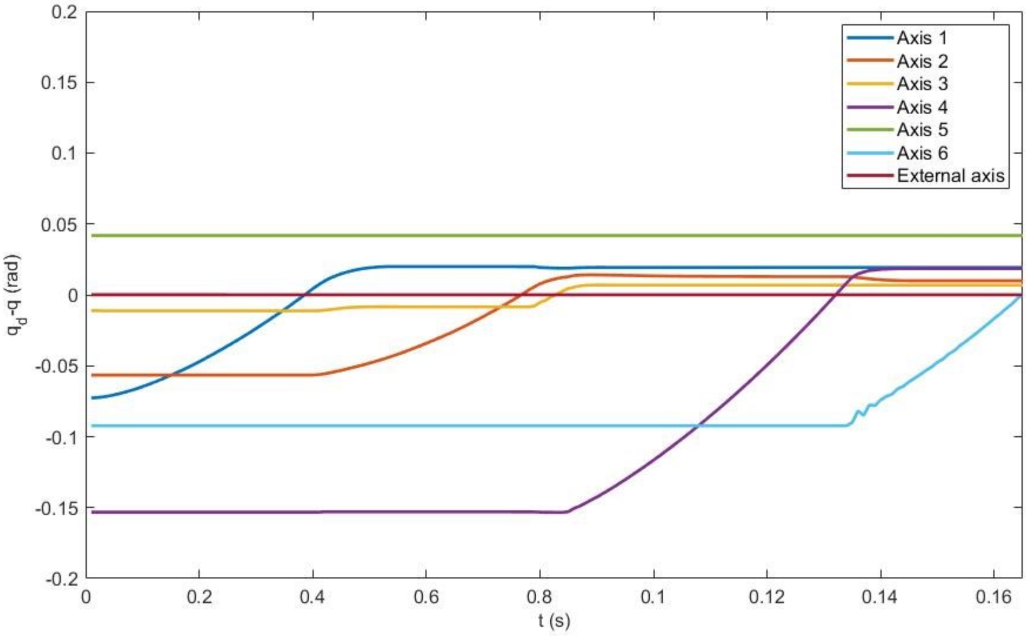



Figure 43: Position error.

### 5.4.4 Analysis and Summary

In 5.4, the combination of SMC, bio-inspired learning and RBF-NN was suggested for trajectory tracking in the presence of switching constraints by robotic manipulators with unknown parameters. The three elements ensure

1) robustness in case of nonlinearities and uncertainties,
2) adaptability through learning in case of changes in the environment and
3) approximation of the nonlinear robot function in case of unknown robot models.

Compared to the BELBIC-SMC-NN-algorithm developed in 5.3, the dependence on fixed variables which need to be initialized a priori was relieved. From the literature survey on RBF-NNs for function approximation in 2.3, need for improvement was identified in the area of initialization and adaptability of variables. The introduction of adaptive NN-parameters: weights, radial basis function centres and widths as well as the interconnection of the three elements: SMC, bio-inspired learning and RBF-NN have shown to remediate the shortcomings of previously suggested concepts.

In 5.4.1, the algorithm was theoretically developed including a Lyapunov analysis.

For preliminary validation purposes, a numerical simulation was performed in a Matlab/Simulink-environment. The simulation (5.4.2) verifies the tracking performance of the controller for the chosen case despite the lack of knowledge of the robot parameters. The black solid line, the desired position is aligned with the dotted blue line, the tracked position. Compliant to the requirements set in 5.1.1, inputs for the simulated trajectory tracking application are lists of successive desired joint positions (.csv-files). The parameter-values for the algorithm were selected small and simple to demonstrate the controller's adaptability as well as its performance independently of specific parameter-values.

As one of the requirements for the developed controller was facilitated practical implementation, the main validation was performed on the KUKA LWR4+-robot arm. The setup for the experimental validation as described in 4.2 was used. Different use cases involving uni- and pluriaxial movements as well as movements in free space, in contact situations and switching between both were considered. As for the simulation, inputs were .csv-files, i.e. lists of successive desired joint positions. The convergence to 0 of the error-signals is graphically presented, i.e. the difference between current joint angular position and desired position converges to 0 over time for all seven robot-joints. This proves that the desired position is reached and that the performance of the control concept is satisfactory.

With the presented algorithm it is possible to address the control problem of following paths by robotic manipulators with unknown models in the presence of switching constraints and uncertainties based on lists of successive desired angular joint positions. The requirements established in 5.1.1 regarding minimization of mean position errors are fulfilled.

## 5.5 Discussion and Summary

The fifth chapter addressed the position control problem in trajectory tracking of switched nonlinear systems. When discontinuously following a succession of curved and straight surfaces, the resulting switching constraints impede the robust performance of the system. Internal and external uncertainties further affect the stability of a controller. The requirements involve the diminution of the difference between actual and desired angular position with a focus on the minimization of the mean joint position error. The design and validation processes presented in Figure 21 were followed for the suggested algorithms.

A theoretical development including a Lyapunov stability analysis sets the necessary analytical basis for the suggested control concepts. A twofold validation process follows. The first validations of the suggested control algorithms are performed in the Matlab/Simulink-environment on a planar RR 2-link-robot arm. In a second step, the algorithms are validated through experimental validations on a 7link KUKA LWR 4+-robot arm. Path following applications with various input densities and switching between free-space and in-contact positions are investigated. Desired positions are partly entered as input in form of tables with successive desired signals, as .csv-files, both in simulation and experiments to make sure the concept is compatible with kinaesthetic teaching. Indeed, in the case of Programming by Demonstration, the signals are recorded as .csv-files. Comparisons with conventional controllers show the outperformance of the new algorithms. PID- and Computed Joint Torque-controllers are chosen because they are, despite their simplicity still widely used in industrial applications and represent both model-free and model-based basic control strategies (Unbehauen, 2008). Linking back to the requirements, the improvements qua maximal, minimal and mainly mean joint position error are proven.

The hypothesis postulated that combining a robust controller with a bio-inspired learning element would merge the constituents' assets while balancing out their respective downsides. To guarantee stability and invariance with respect to uncertainties for the considered nonlinear system, sliding mode control was chosen as robust controller. A reinforcement learning based element was chosen to ensure adaptability to changes in the environment, to switching constraints and uncertainties or disturbances. In addition, the learning extension would alleviate the drawbacks of the robust controller (chattering and required a priori knowledge). The absolute mean position error was kept below the targeted threshold of 0.3 rad and the approximated robot function did not affect this.

SMC was combined with Brain Emotional Learning Based Intelligent Control, BELBIC. The concept modelling the mammalian emotional learning behaviour via the interplay of amygdala and orbitofrontal cortex in the limbic system, had been introduced by (Lucas, 2004) in the beginning of 21st century and was chosen here because of its learning capacity and intuitiveness as it acts like a very abstracted and simplified version of the human brain. The algorithm was extended for the case of unknown robot dynamics. The latter were estimated with an artificial neural network. The dynamic parameters are summarized in a single nonlinear function in order to reduce the estimation burden and error. Instead of estimating the unknown parameters $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ separately, they are approximated grouped in a nonlinear robot-function $\boldsymbol{f}$. A simple perceptron model without training phase meets the needs for function-approximation in the considered use case. A RBF-NN was chosen for the approximation as it remains in the general idea of bio-inspiration of the work and has shown optimal approximation performance. 5.4 facilitates the implementation of the control concept by releasing the dependence on a priori determined fixed parameters. As far as the estimation of the robot function is concerned, the implementation of adaptive laws for updating not only the weights of the RBF-NN, but also the centres and widths of the function allows the optimal approximation of unknown functions. As far as the learning element is concerned, the adaptive motivation-lifecycles of the human learning behaviour inspired the element which replaces BELBIC. This amendment follows up on the improved intuitiveness and increased implementation and usability options.

This concludes the position control-chapter. As for most processes trajectory tracking or pure position control is not sufficient, torque control is added in the next step. The application of specified forces at particular locations on the tracked trajectory is desired for a variety of use cases. As the simple playback of recorded or desired torques results in unstable behaviour and exhibits significant chattering, the implementation of a controller is necessary (Lee, 2015). Therefore chapter 6 addresses combined position and force control.

# 6 Combined Position and Force Control

The sixth chapter is dedicated to the second sub-challenge as identified in 1.1.4, namely path following + application of a force, combined position and force control. The addition of a specified force during a trajectory tracking operation is necessary for successfully carrying out a variety of contact-based tasks. As pointed out in the first chapter, the execution of such a process is twofold. It requires

1) the following of the surface to ensure the constant contact between robot end-effector or tool and workpiece and

2) the application of an adequate force or in angular form: torque.

For these processes, manual work is current state-of-the-art. The fact that these processes were designed by and for humans makes them the most appropriate performers for these complex tasks. Additionally, human capabilities are well suited for these tasks. The challenge in automation is to get inspired by the approach of the human to perform the considered task by translating his capabilities into robot skills and by including the expertise of the human in the control algorithm.

The purposes of this chapter are first to define the requirements for the combined position and force control problem and second to develop and validate appropriate control algorithms. Figure 45 graphically represents the structure of the sixth chapter.

The first part, 6.1, describes the combined position and force control challenge. First, the problem is formulated. Second, the requirements are deduced from the research question (chapter 3), the application area (chapter 1) and the gaps identified in the state-of-the-art (chapter 2). Third, the modus operandi is introduced.

The procedure of parts 6.2 and 6.3 can be described as top-down. 6.2 is based on the model-based position control algorithm suggested in 5.4. 6.3 suggests a compressed model-free control algorithm focusing on the most essential elements of the complex controllers analysed before.

In 6.4, the results of the sixth chapter are summarized and discussed.

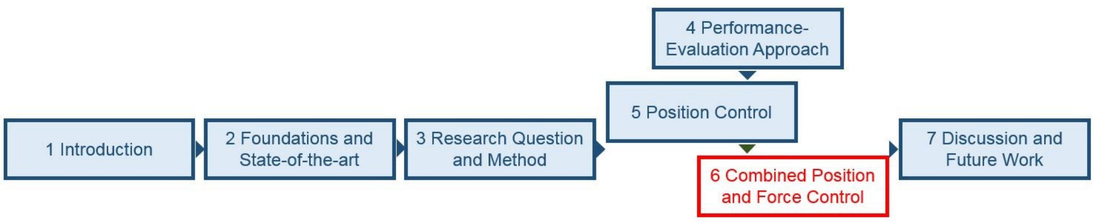



Figure 44: Situation of chapter 6 Combined Position and Force Control.

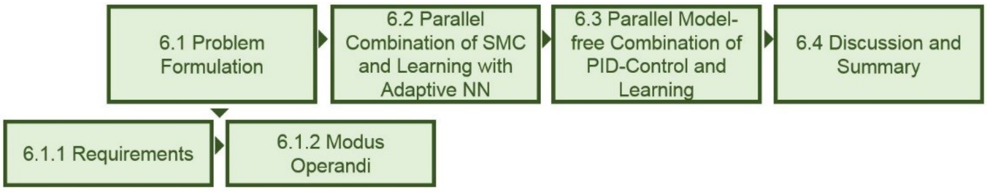



Figure 45: Structure of chapter 6 Combined Position and Force Control.

## 6.1 Problem Formulation

This part provides a detailed description of the problem addressed here. In contrast to the rather general formulation of problem and objectives following the identification of the scientific gap from the literature survey, this problem formulation is more specific and dedicated exclusively to the second sub-challenge. The aim is to combine freeform trajectory tracking with force control and to develop a control strategy which enables an industrial robot arm to follow a desired freeform-path and simultaneously apply specified adequate joint-torques at the appropriate moment and position. Additionally, the effects of friction are taken into account.

The considered application is a human-centred process, i.e. it was designed by and for humans and their capabilities. Automation of this type of process therefore asks for inspiration by the humans' approach to perform the task through translating their capabilities into robot skills and taking advantage of the workers' expertise in the control algorithm.

The robot arm which has been considered in chapter 5 is extended in this part of the work as in eq. (82).

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{d} + \boldsymbol{r} + \boldsymbol{Q}_i + \boldsymbol{u} \tag{82}$$

where $\boldsymbol{r} \in R^n$ stands for the friction between end-effector and environment or surface. The friction is a function of the applied torque $\boldsymbol{\tau} \in R^n$ and the robot link velocity: $\boldsymbol{r} = \boldsymbol{\gamma}\boldsymbol{\tau}_d{}^T\dot{\boldsymbol{q}}_{actual}$ with $\boldsymbol{\gamma}$ a constant vector. $\boldsymbol{u} \in R^n$, the applied torque is here the sum of the outputs of the pose-controller and the force-controller, i.e. $\boldsymbol{u}_q + \boldsymbol{u}_\tau$. The control action consists in adapting both robot joint positions and the applied forces to match the desired poses and forces. As the simple playback of recorded or desired torques results in unstable behaviour and exhibits significant chattering, the implementation of an additional controller becomes necessary (Lee, 2015).

The desired signals in turn are put as .csv-files into the system. These lists of successive joint angles and torques are obtained from the records of a human performing the considered task operating the lightweight robot arm in gravity compensation mode.

### 6.1.1 Requirements

The suggested control algorithms address path following with application of specified torques on similar freeform-workpieces with known, constant positioning. The aim is to develop bio-inspired algorithms for combined position and torque control in an automated trajectory tracking process mimicking the human approach.

The objective is to keep the error between desired and measured signals minimal at all times. The absolute mean errors of the torque- and position-signals have to be minimized. For the position error the threshold of 0.3 rad established for the position controller is maintained. The threshold for the absolute mean torque error is fixed at 0.1 Nm. The values are deduced based on the performance of PID-control and results presented by the scientific community (Jasim, 2015) (Xu, 2007).

For the case of unknown robot dynamic parameters, the aim is to either approximate them in the form of the nonlinear robot function $\boldsymbol{f}$ (eq. (9)) or use a model-free concept. Performance in terms of position and torque error requirements is invariant to unknown robot dynamics.

Following the vision of a human mimicking concept with facilitated implementation and use, some additional requirements have to be considered. First, the input for the controller consists of lists of desired successive joint-specific positions and torques obtained from the records of a human performing the considered task operating the lightweight robot arm in gravity compensation mode. Compliance with kinaesthetic teaching and PbD is required. Second, the suggested concept should renounce the use of external sensors, no cameras or force-torque sensors are added. Third, for an implementation in industry and reduced computational burden, the most simple possible control concept is desired. In this sense a balance is to be found between a too simple concept with poor performance and a too complex concept with optimized performance.

### 6.1.2 Modus Operandi

6.2 and 6.3 follow the same outline as in chapter 5 and depicted in Figure 46.

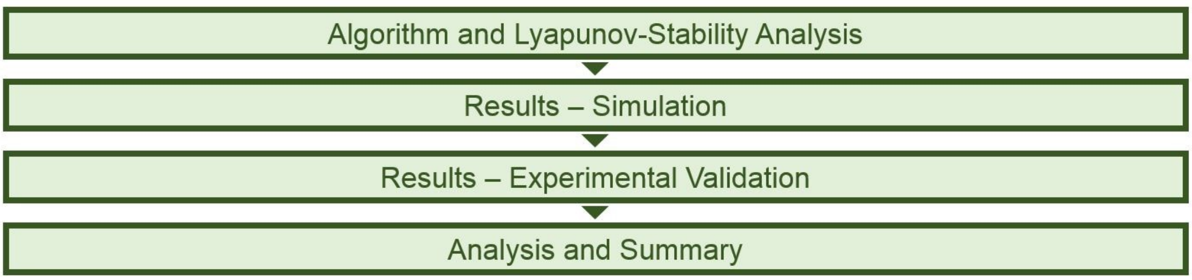



Figure 46: Modus Operandi.

## 6.2 Parallel Combination of SMC and Learning with Adaptive NN

In 5.4, the position control performance of the combination of robust SMC, bio-inspired learning element and adaptive RBF-NN was shown. The controller achieved good results even in the presence of switching constraints, uncertainties and unknown robot model parameters. Pure position control, however is insufficient to address contact-rich tasks and to control the application of desired forces (Abu-Dakka, 2015) (Kronander, 2014) (Siciliano, 2008). Because pure force control is not sufficient either due to contact instabilities, combined position and force control needs to be ensured (Newman, 1999) (Phillips, 2016). Based on the literature survey in 2.4, it is hypothesized that a parallel control concept is able to address combined path following + application of a force. Parallel position and force control ensures

1) following the desired trajectory and

2) applying an adequate force or torque at specified positions (as recorded during experiments).

In this section, the ideas developed in 5.4 are adopted and transferred to a parallel position and force control concept.

### 6.2.1 Algorithm and Lyapunov-Stability Analysis

In this section, the combination of robust SMC, bio-inspired learning element and adaptive RBF-NN as developed in 5.4 is adopted for parallel position and torque control. Two control outputs, $\boldsymbol{u}_q \in R^n$ and $\boldsymbol{u}_\tau \in R^n$ act in parallel to address position and torque control, respectively.

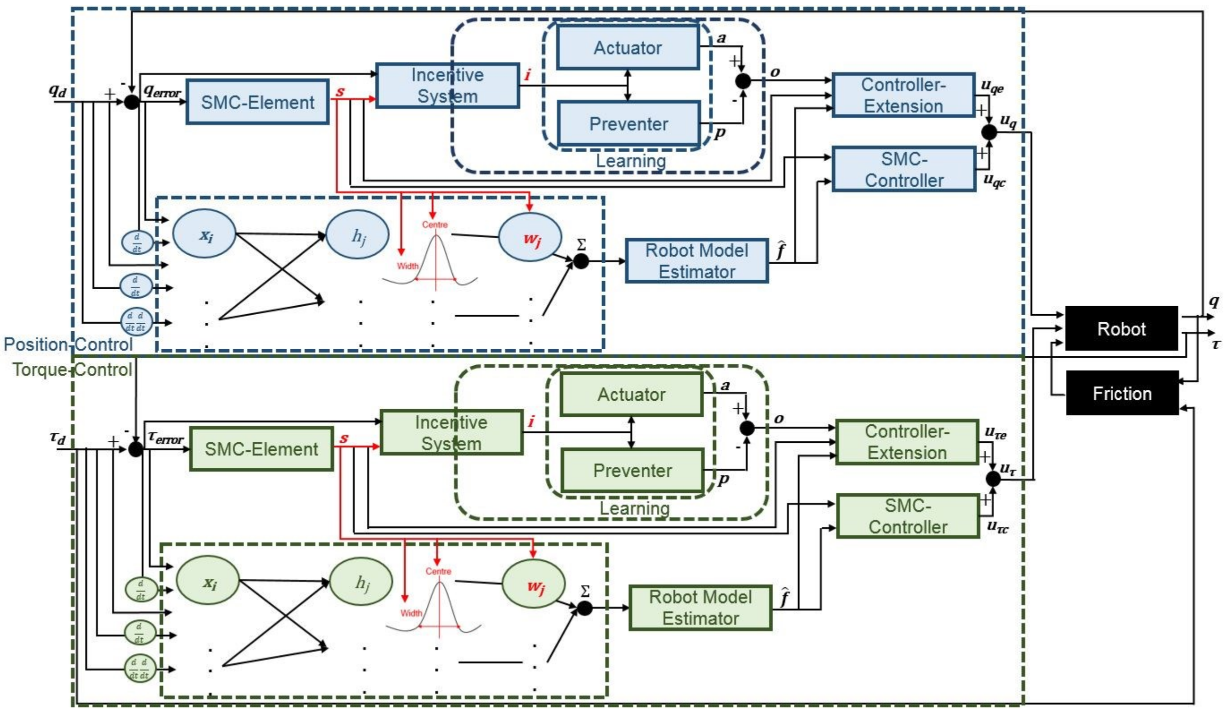



Figure 47: Block diagram of the control concept.

Both outputs are composed of a conventional controller and a bio-inspired learning controller-extension. $\boldsymbol{u}_q \in R^n$ follows the equations developed in 5.4.1 while $\boldsymbol{u}_\tau \in R^n$ follows the same equations using $\boldsymbol{\tau}$ instead of $\boldsymbol{q}$ and with $\boldsymbol{\tau_{measured}} = \boldsymbol{u}_\tau + \boldsymbol{r}$. The Lyapunov-stability analysis can therefore be adopted from 5.4.

Figure 47 illustrates the parallel position and force control concept including the interconnection of robust SMC, bio-inspired learning and adaptive RBF-NN as discussed in 5.4.

### 6.2.2 Results - Simulation

As a first validation of the suggested parallel control concept, trajectory tracking with application of torques is simulated in a Matlab/Simulink-environment. The details of the robotic manipulator can be found in 4.1 and $\boldsymbol{\gamma} = 1\boldsymbol{I_2}, D_{pq} = 5, D_{iq} = 20, D_{dq} = 20, D_{p\tau} = -5, D_{i\tau} = -35, D_{d\tau} = 0$. The inputs are .csv-files, i.e. lists of successive desired joint positions in radians and joint torques in Nm, respectively.

The parameter-values were selected small and simple as in 5.4.2 to demonstrate the controller-performance independently of specific parameter-values.

The simulation results are represented in Figures 48-51. Figure 48 shows the angular position tracking for both robot links. Figure 50 depicts the torque tracking.

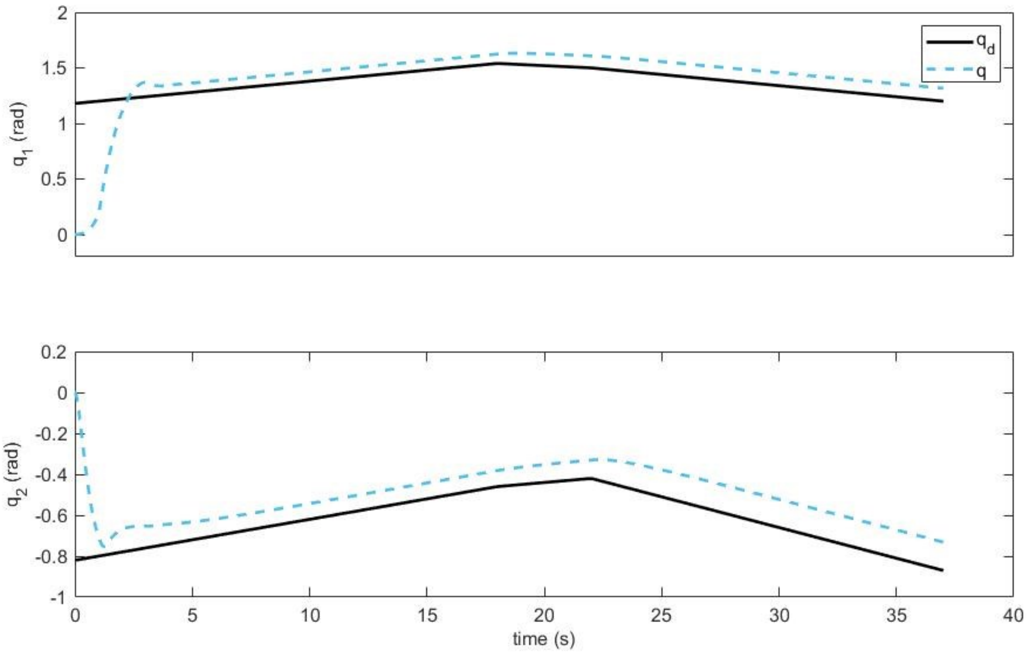



Figure 48: Position tracking of link 1 (top) and link 2 (bottom).

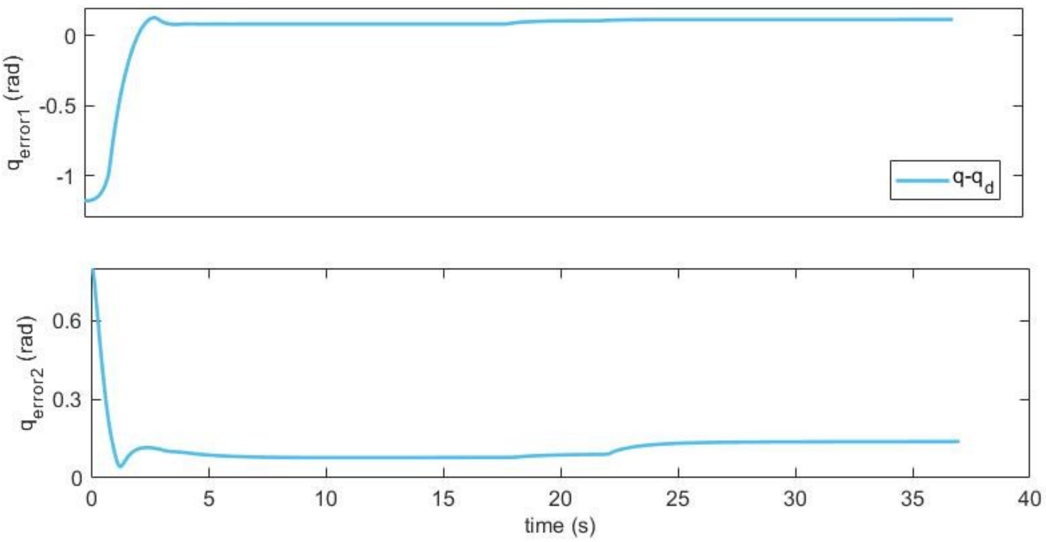



Figure 49: Position error of link 1 (top) and link 2 (bottom).

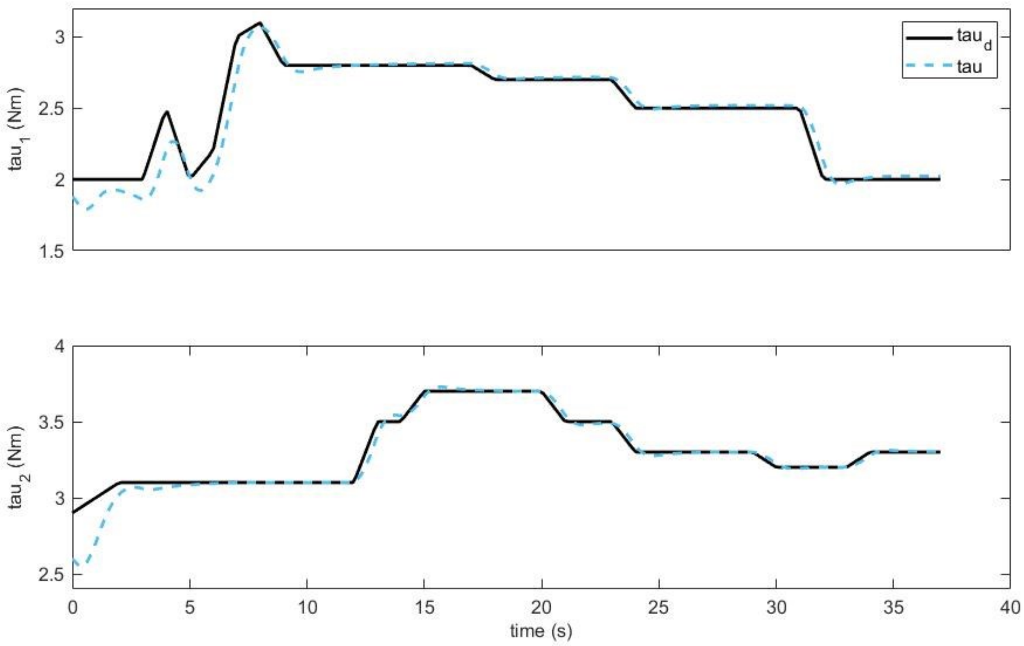



Figure 50: Torque tracking of link 1 (top) and link 2 (bottom).

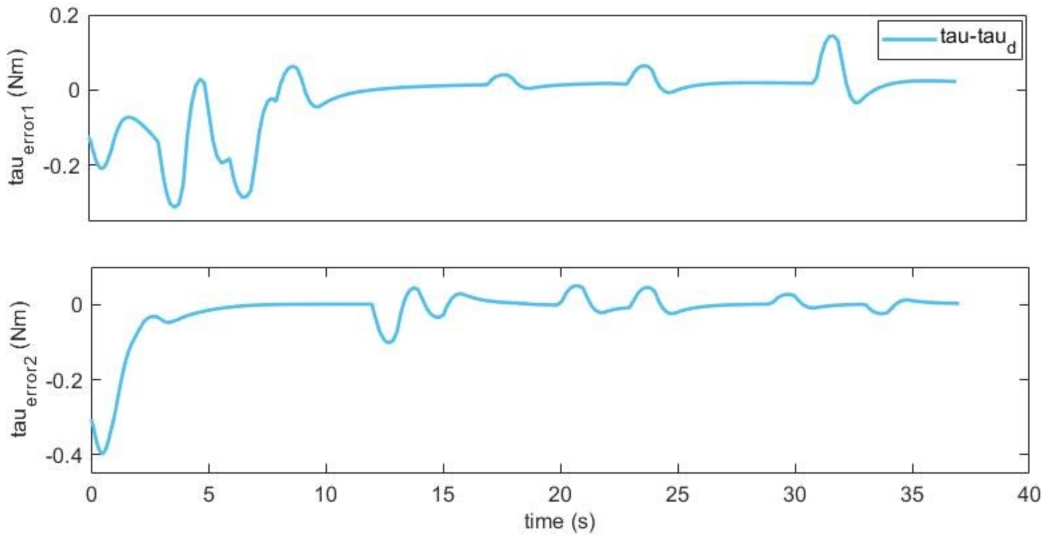



Figure 51: Torque error of link 1 (top) and link 2 (bottom).

The maximal, minimal and mean error-values of positions and torques of all manipulator-links are computed for the suggested control scheme as well as for a parallel controller comprising a PID-position controller and a PID-torque controller (Table 5).

Table 5: Absolute maximal, minimal and mean tracking errors for both manipulator-links using the suggested concept and a parallel controller-combination of PID-elements.

| | **suggested** | **concept** | **PID** | |
|---|---|---|---|---|
| | $q_{error,1}$ | $q_{error,2}$ | $q_{error,1}$ | $q_{error,2}$ |
| **Position** | | | | |
| Max [rad] | 1.18 | 0.82 | 3.1594 | 0.82 |
| Min [rad] | 0.004 | 0.0423 | 9.6971e-04 | 2.246e-05 |
| Mean [rad] | 0.274 | 0.1767 | 1.6392 | 0.0867 |
| **Torque** | | | | |
| Max [Nm] | 0.3118 | 0.3981 | 2.4114 | 2.4497 |
| Min [Nm] | 1.8887e-05 | 4.2338e-05 | 0.0026 | 0.0013 |
| Mean [Nm] | 0.0643 | 0.0679 | 1.5595 | 1.616 |

### 6.2.3 Results – Experimental Validation

The experimental validation is done on the KUKA LWR 4+-robot arm which was described in 4.2.

The motion sequence of the experiment is depicted in Figure 52: from free space ($1^{st}$ row left) the tool is brought into contact with the workpiece surface ($1^{st}$ row right), follows a straight line ($2^{nd}$ row left) and finally applies a force on the surface ($2^{nd}$ row right). Figures 53 and 54 show the evolution of the error signals of angular position and torque, respectively. Both show convergence to 0.

Figure 52: End-effector movement and behaviour.

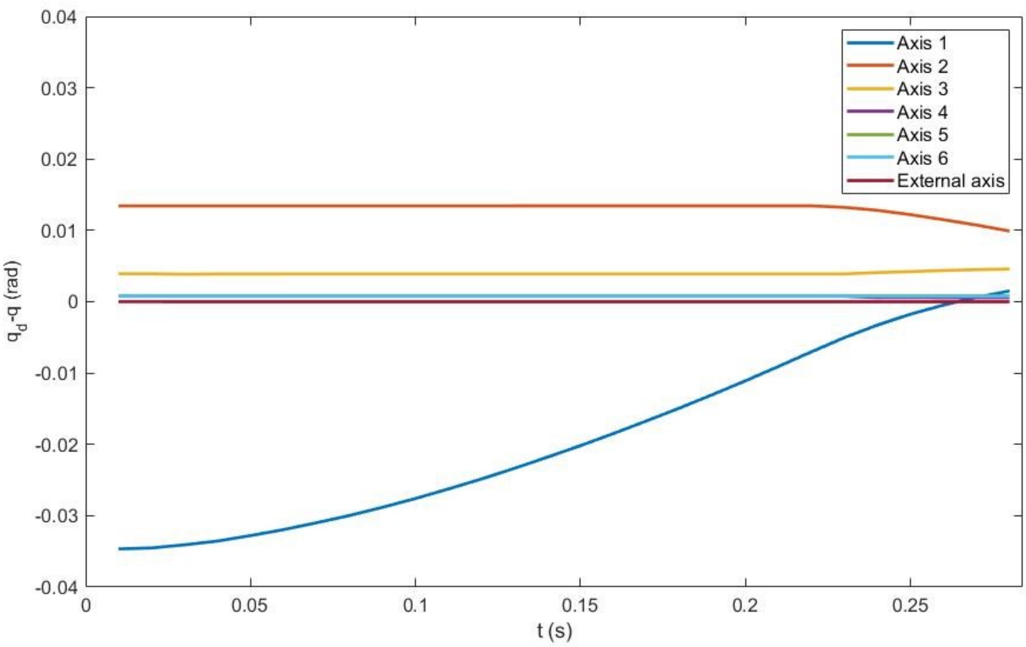



Figure 53: Position error.

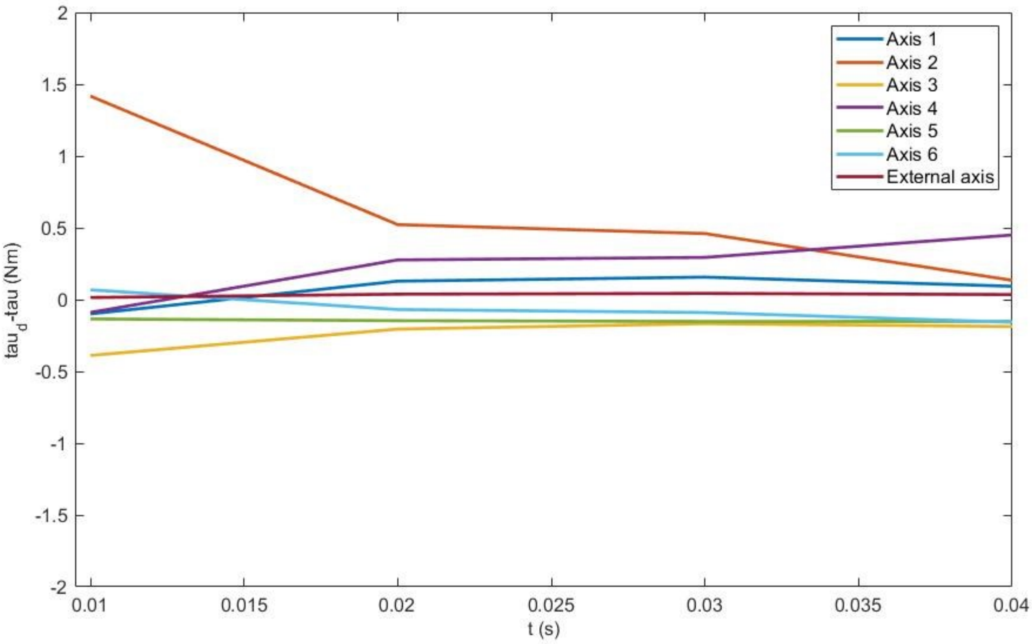



Figure 54: Torque error.

### 6.2.4 Analysis and Summary

The combined position and force/torque control problem as found in constrained trajectory tracking, in path following with application of a specified force/torque, was addressed. Combinations of robust SMC, bio-inspired learning and adaptive RBF-NN were implemented to control position and torque, respectively. The three elements ensure

1) robustness in case of nonlinearities and uncertainties,
2) adaptability through learning in case of changes in the environment and
3) approximation of the nonlinear robot function in case of unknown robot models.

From the literature survey, it was derived that position and torque control need to be combined to successfully address constrained trajectory tracking and that a parallel concept is optimally suited. The suggested parallel controller uses the concept developed in 5.4. The hypothesis is that the concept which was successfully implemented for position control will also be performing well for parallel position and torque control.

6.2.1 is dedicated to the theoretical development of the parallel concept and to the references to 5.4.1.

As a first step in the validation process, a numerical simulation was performed in the Matlab/Simulink-environment. For as far as the simulation is concerned (6.2.2) it verifies the position and torque tracking performance of the parallel controller. There is a good correspondence between the solid black line representing the desired position and the dotted blue line representing the tracking position. The same observation can be made for the correspondence between the desired torques (solid black lines) and the tracking torques (dotted blue lines). Compliant to the requirements set in 6.1.1, inputs for the simulated application are .csv-files, i.e. lists of successive desired joint positions or torques, respectively.

The parameter-values for the algorithm were consciously and arbitrarily selected small and simple as in 5.4.2 to demonstrate the controller's adaptability as well as its performance independently of specific parameter-values.

The subsequent validation step is an experimental proof on the KUKA LWR4+-setup as described in 4.2. A use case involving path following and application of specified torques was analysed. As for the simulation, inputs were .csv-files, i.e. lists of successive desired joint positions and torques. The convergence to 0 of the error-signals is graphically presented, i.e. the differences between current joint angular position and desired position or current joint torque and specified torque, respectively converge to 0 over time for all relevant joints. This proves that the desired values for positions and torques are reached and that the parallel control concept is performing as expected.

With the presented algorithm it is possible to address the combined position and torque control problem of constrained trajectory tracking by robotic manipulators with unknown models in the presence of switching constraints and uncertainties based on lists of successive desired angular joint positions and torques. The requirements established in 6.1.1 regarding minimization of mean errors are fulfilled. Absolute mean position errors are kept below 0.3 rad. Torque errors meet the requirement of <0.1 Nm, presenting not even a tenth of the value computed for the PID-control concept.

As far as laboratory settings are concerned, the suggested control concept is appropriate, i.e. it assures of its performance in research-focused environments. As far as industrial environments are concerned, the simpler and more straightforward the controller the better it is suited for implementation. In this context, the complexity of the suggested control concept can be considered a downside. In order to facilitate future industrial use, the aim is to simplify the controller through focussing on its most essential elements. The latter were identified through the analyses of the previous chapters. In the following section, a balance has to be found between complex, highly performing and simple, poorly performing controller. The finally suggested concept will be a trade-off between the concept presented in 6.2 and a conventional PID-controller.

## 6.3 Parallel Model-free Combination of PID-Control and Learning

In 6.2, the hypothesis that a parallel concept with two controller-halves each combining SMC for robustness and learning for flexibility with adaptive RBF-NN for estimating unknown parameters can successfully address a combined position and torque control problem was corroborated. In research-focused settings, the suggested complex model-based concept shows optimal results via simulation and experiments. In industrial environments, priorities however are different: here simple design and straightforward implementation are fundamental. For an implementation in industry, the concept suggested in 6.2 needs to be simplified. Major drawbacks of model-based controllers are their complex design and implementation processes (Visser, 2010). Therefore, in this section a model-free concept is developed which focuses only on the essential elements of the complex algorithm from 6.2.

The aim is to find a balance between the complex model-based but optimally performing algorithm developed in 6.2 and a simple model-free but unsatisfactorily performing controller.

### 6.3.1 Algorithm

A parallel control concept involving two model-free controller-halves for position and torque control, respectively is developed. Inspired by the algorithm suggested in 5.2, 5.3, 5.4, 6.2 both independent controller-halves combine a conventional controller with a bio-inspired learning element. A PID-controller as described in 2.1.2 is chosen as conventional controller because it is widely used in industry (Adar, 2016) (Desborough, 2002) (Unbehauen, 2008). The learning element is similar to the reinforcement learning-based concept developed and used in 5.4 and 6.2. It uses an incentive and a learning system as well as the interaction of an actuator and a preventer. An incentive system transforms sensory information into an incentive, i.e. a reward-based extrinsic motivational stimulus. Depending on the environment, the stimulus to the agent, i.e. how to maximize the reward for the system is changed. This adaptive incentive then forms the input to a learning system which feeds both an actuator and a preventer. The interplay of the latter is inspired on the interplay of the amygdala and the orbitofrontal cortex in the mammalian brain during emotional learning. While the actuator establishes stimulus-action associations, the preventer erases associations which are no longer needed. The removal of no longer relevant stimulus-action associations is essential for a successful learning and to reduce the amount of data in the system. The latter is similar to the phenomenon of synaptic plasticity in the human brain. The developed concept-idea is graphically represented in Figure 55. Despite the concept being inspired by the functioning of the human brain, it does not attempt to accurately model its structure. Rather than presenting a true-to-life computational model of the mammalian learning behaviour, the aim is to improve conventional PID-control through the implementation of neuro-inspired concepts.

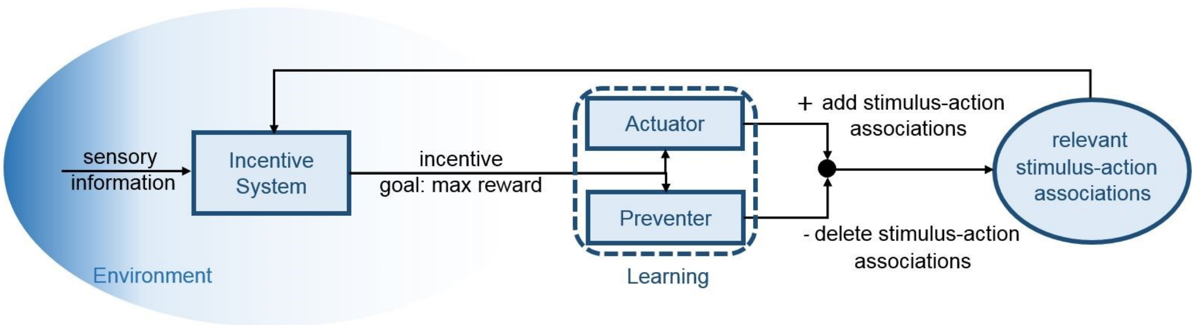



Figure 55: The principle of the learning concept.

Hereafter, the position- as well as the torque-controller-halves are developed.

Position Control

The conventional PID-controller-output as in 2.1.2 is repeated in eq. (83)

$$\boldsymbol{u}_{q_{conventional}} = -\boldsymbol{D}_{pq}\boldsymbol{q}_{error} - \boldsymbol{D}_{iq}\int \boldsymbol{q}_{error}\ dt - \boldsymbol{D}_{dq}\dot{\boldsymbol{q}}_{error} \tag{83}$$

where $\boldsymbol{D}_{pq}$, $\boldsymbol{D}_{iq}$ and $\boldsymbol{D}_{dq}$ are constant gain factors and with the error-signal, i.e. the difference between measured and desired signal (eq. (3)).

For the learning element, the value of the sensed state is defined as the error-signal (eq. (84)). As the only way to collect information about the environment is to interact with it, a feedback-loop is implemented in this controller-part.

$$\boldsymbol{w_{state}} = \boldsymbol{q} - \boldsymbol{q_d} = \boldsymbol{q_{error}} \tag{84}$$

The reward, i.e. incentive $\boldsymbol{i} \in R^{\boldsymbol{n}}$ is defined in eq. (85)

$$i_j = sgn\left(w_{state_j}\right)\dot{w}_{state_j}(w_{state_j} - o_j) \tag{85}$$

with $j = 1, \dots n$ indicating the considered link.

The incentive is the input to the learning system which is composed of the interaction between an actuator and an inhibitor. The outputs and learning rates of both, the actuator and the preventer are given in eq. (86)-(89).

$$\boldsymbol{a} = \alpha\boldsymbol{w_{state}}.*\ \Delta_a \tag{86}$$

$$\boldsymbol{p} = \alpha\boldsymbol{w_{state}}\ .*\ \Delta_p \tag{87}$$

$$\Delta_{a_j} = \alpha\dot{w}_{state_j}\max\left(0, i_j\right) \tag{88}$$

$$\Delta_p = \alpha\dot{\boldsymbol{w}}_{\boldsymbol{state}}.*\ (\boldsymbol{o} - \boldsymbol{i}) \tag{89}$$

with $\alpha$ a constant factor.

The interplay of actuator and preventer results in output $\boldsymbol{o} \in R^n$ which subtracts the preventer-output from the actuator-output (eq. (90)). This guarantees only relevant connections are kept. Mimicking synaptic plasticity, this law allows to limit the number of active learned connections.

$$\boldsymbol{o} = \boldsymbol{a} - \boldsymbol{p} \tag{90}$$

The controller-extension-output is defined in eq. (91), the integration over time mimicking experience.

$$\boldsymbol{u}_{\boldsymbol{q_{extension}}} = \beta\boldsymbol{w_{state}} - \int_0^T \boldsymbol{o}\,(t)dt \tag{91}$$

with $\beta$ being a constant gain-factor.

The final position-controller output combines the outputs of the conventional controller and of the extension.

$$\boldsymbol{u_q} = \boldsymbol{u}_{\boldsymbol{q}_{conventional}} - \boldsymbol{u}_{\boldsymbol{q_{extension}}} \tag{92}$$

Torque Control

Parallel to the position-controller, a torque-controller is implemented to make sure the desired forces/torques are applied.

The conventional PID-controller-output is defined in eq. (93)

$$\boldsymbol{u}_{\tau_{conventional}} = -\boldsymbol{D}_{p\tau}\boldsymbol{\tau_{error}} - \boldsymbol{D}_{i\tau}\int \boldsymbol{\tau_{error}} - \boldsymbol{D}_{d\tau}\dot{\boldsymbol{\tau}}_{\boldsymbol{error}} \tag{93}$$

where $\boldsymbol{D}_{p\tau}$, $\boldsymbol{D}_{i\tau}$ and $\boldsymbol{D}_{d\tau}$ are constant gain-matrices and the error-signal

$$\boldsymbol{\tau_{error}} = \boldsymbol{\tau_{measured}} - \boldsymbol{\tau_d} \tag{94}$$

with $\boldsymbol{\tau_{measured}} = \boldsymbol{u}_\tau + \boldsymbol{r}$.

For the learning controller-extension, the state-value $\boldsymbol{w}_{state}$ is defined as the error-signal. Further eq. (85)-(90) apply. The controller-extension-output follows eq. (95).

$$\boldsymbol{u}_{\tau_{extension}} = -\beta \boldsymbol{w}_{state} + \int_0^T \boldsymbol{o}(t)dt \quad (95)$$

The final torque-controller output combines the outputs of the conventional controller and of the extension.

$$\boldsymbol{u}_{\tau} = \boldsymbol{u}_{\tau_{conventional}} - \boldsymbol{u}_{\tau_{extension}} \quad (96)$$

The suggested method attempts to combine robustness, simplicity and intuitiveness for combined position and torque control and is depicted in Figure 56. The figure illustrates the combination of position and torque control as well as the interplay of conventional control and learning element.

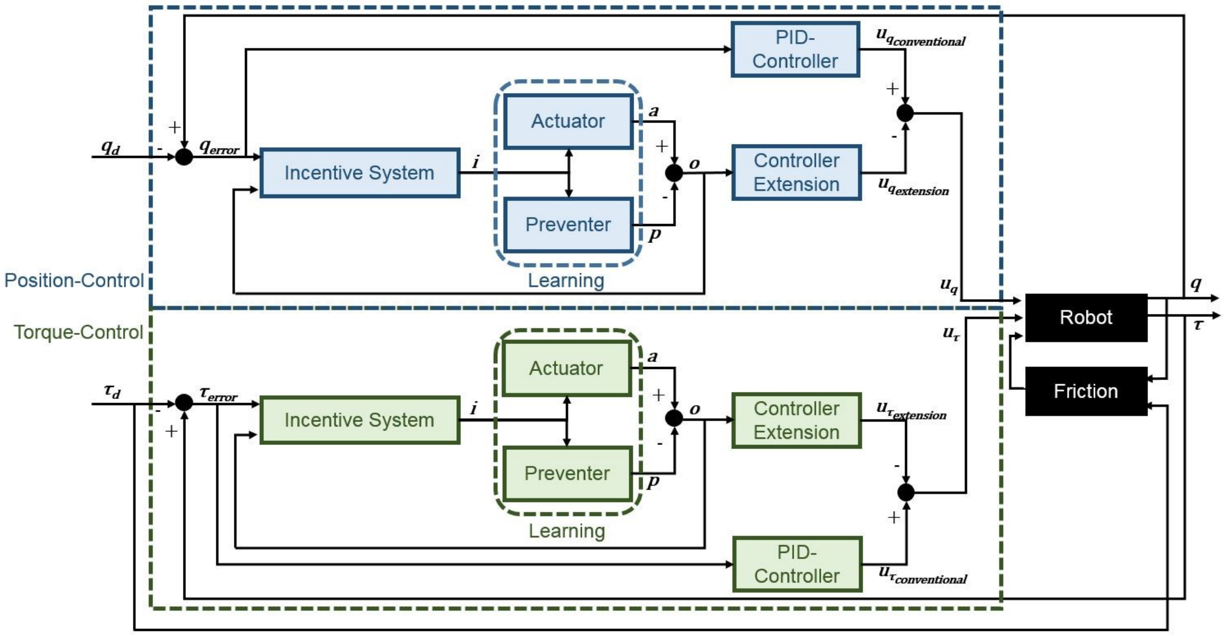



Figure 56: Block diagram of the control concept.

### 6.3.2 Results – Simulation

As a first validation of the suggested concept, a surface finishing process is simulated in an abstracted manner. The application consists in the robot arm successively following desired positions and applying desired torques at specified positions. The desired angular positions and joint-torques, the system-inputs are provided in two .csv-files.

The controller-parameters introduced in eq. (83)-(96) are chosen as follows: $\alpha = 5, \beta = 30, \boldsymbol{\gamma} = 1\boldsymbol{I_2}, D_{pq} = 5, D_{iq} = 20, D_{dq} = 20, D_{p\tau} = -5, D_{i\tau} = -35, D_{d\tau} = 0$. Due to the small amount of parameters and a limited range between ±50, tuning is not excessively time-consuming. For PID-parameter tuning a variety of methods can be found in literature (Cervantes, 2001) (Desborough, 2002) (Siciliano, 2008) (Unbehauen, 2008).

Excellent results were obtained in simulation (Klecker, 2018). The performance-results are illustrated in Figures 57-60. While Figure 57 shows the position-tracking performance of the suggested controller scheme, Figure 59 depicts its torque-tracking results.

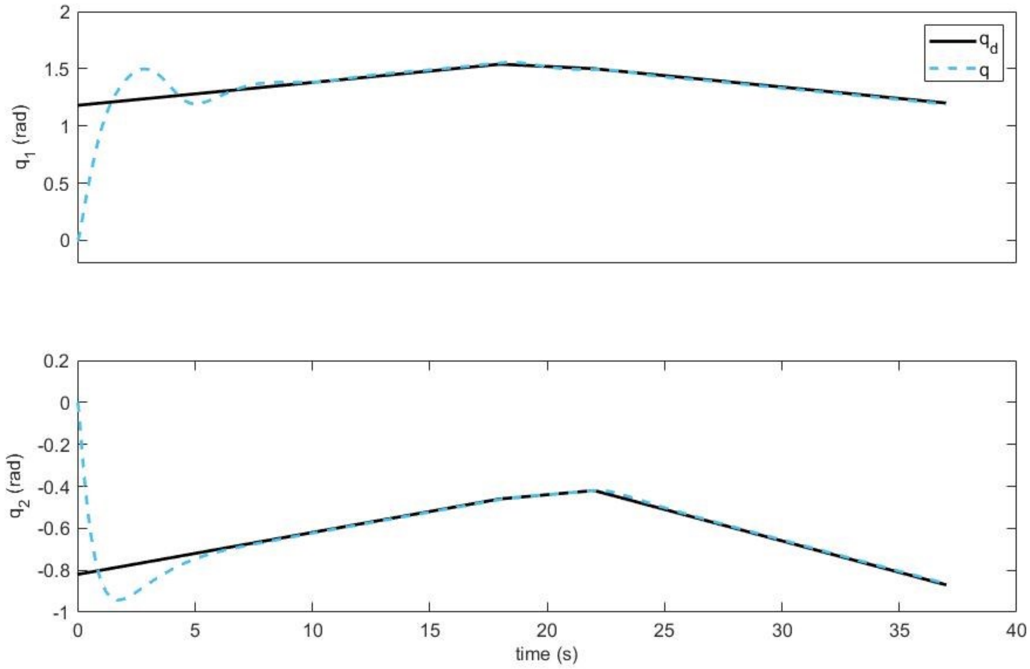



Figure 57: Position tracking of link 1 (top) and link 2 (bottom).

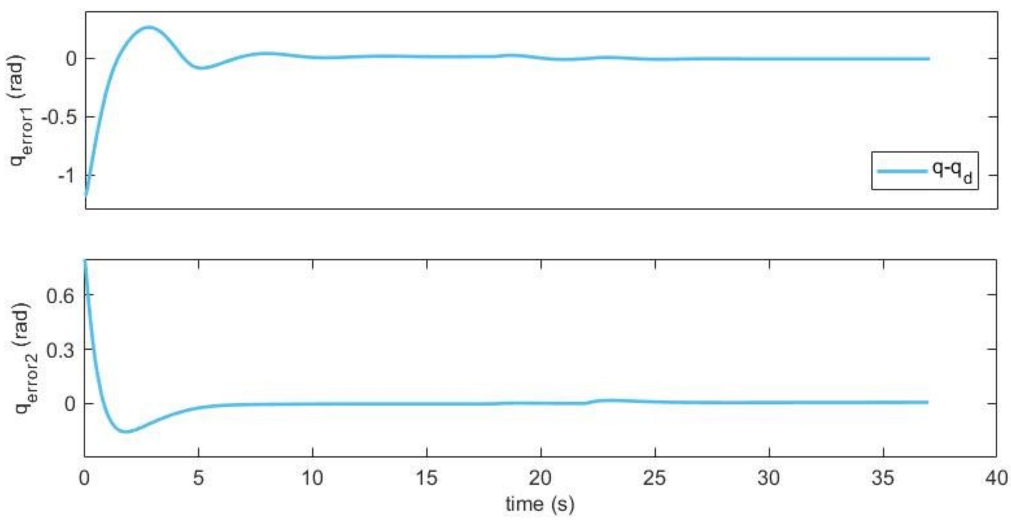



Figure 58: Position error of link 1 (top) and link 2 (bottom).

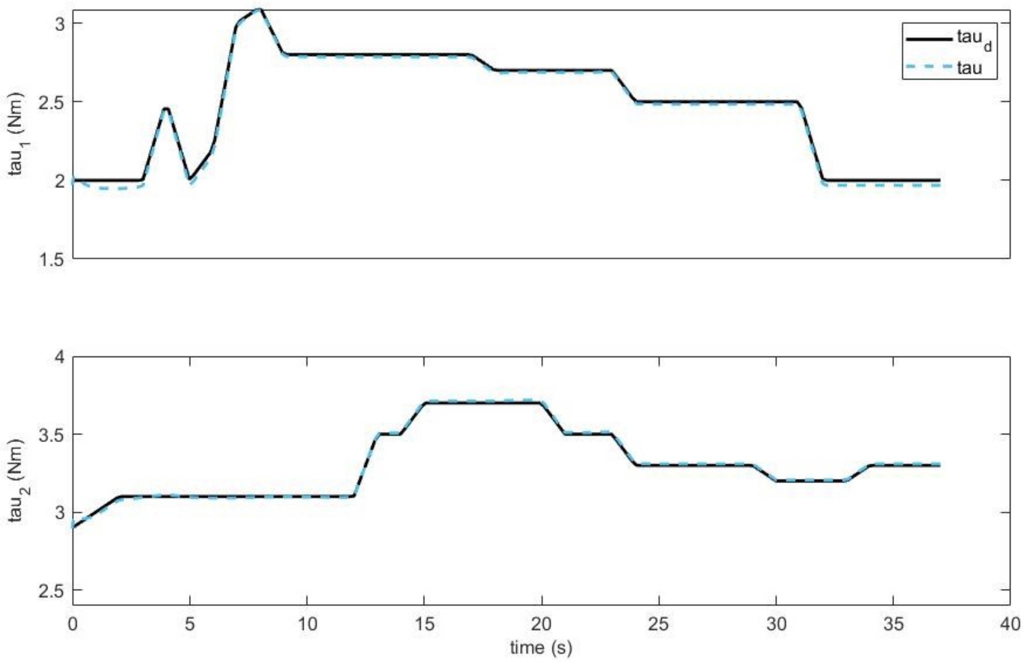



Figure 59: Torque tracking of link 1 (top) and link 2 (bottom).

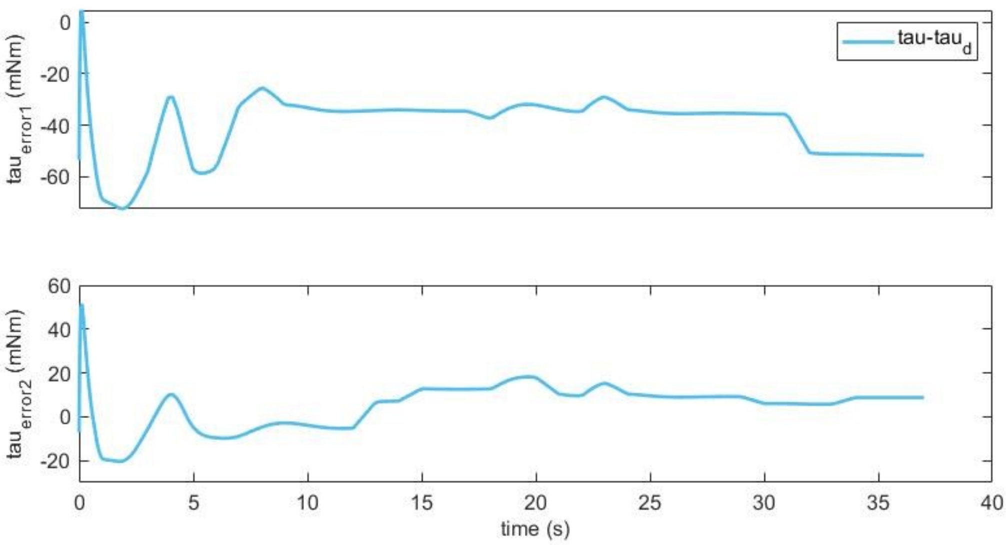



Figure 60: Torque error of link 1 (top) and link 2 (bottom).

Table 6 shows the added value of the extension-elements $\boldsymbol{u}_{q_{extension}}$ and $\boldsymbol{u}_{\tau_{extension}}$. The maximal, minimal and mean error-values of positions and torques of all manipulator-links are computed for the suggested control scheme as well as for a parallel controller comprising a PID-position controller and a PID-torque controller.

Table 6: Absolute maximal, minimal and mean tracking errors for both manipulator-links using the suggested concept and a parallel controller-combination of PID-elements.

| | **suggested** | **concept** | **PID** | |
|---|---|---|---|---|
| | $q_{error,1}$ | $q_{error,2}$ | $q_{error,1}$ | $q_{error,2}$ |
| **Position** | | | | |
| Max [rad] | 1.18 | 0.82 | 3.1594 | 0.82 |
| Min [rad] | 6.2119e-04 | 7.0763e-06 | 9.6971e-04 | 2.246e-05 |
| Mean [rad] | 0.1351 | 0.0806 | 1.6392 | 0.0867 |
| **Torque** | | | | |
| Max [Nm] | 0.0525 | 0.0515 | 2.4114 | 2.4497 |
| Min [Nm] | 0.0042 | 1.4151e-05 | 0.0026 | 0.0013 |
| Mean [Nm] | 0.0203 | 0.011 | 1.5595 | 1.616 |

### 6.3.3 Results – Experimental Validation

For the final validation, the experimental setup described in 4.2 is used. It is subdivided in three phases: before the parallel control of position- and torque-signals is tested, position- and torque-control are validated separately.

<u>Position Control</u>

As a first experiment a position has to be reached. Figure 61 proves that the position error-signals of all 7 joints converge to 0, i.e. that the end-effector reaches its goal position.

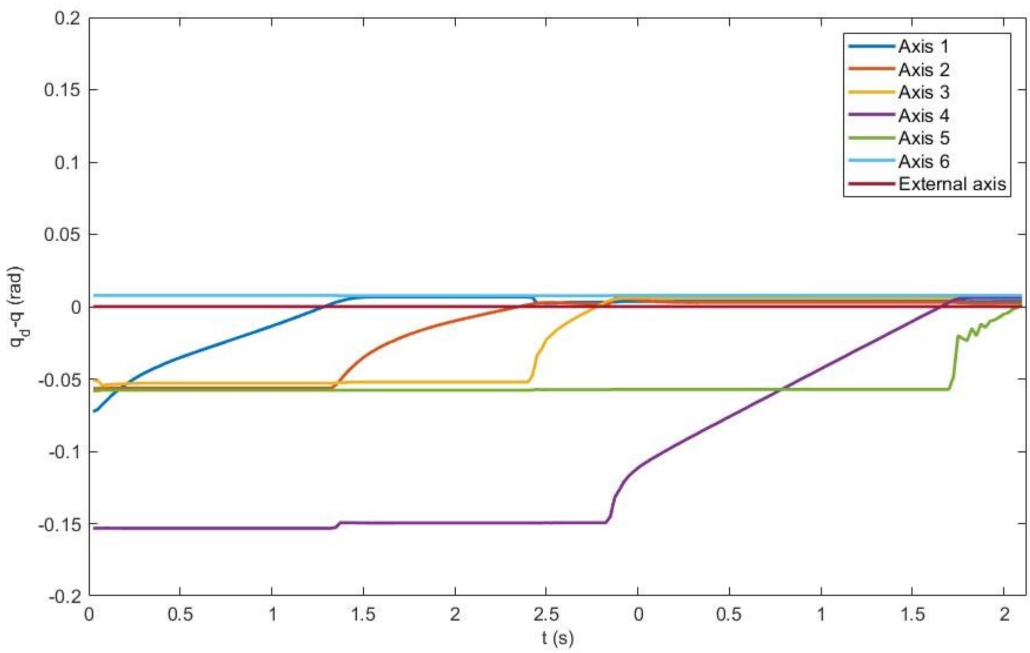



Figure 61: Position error.

The second experiment presents the combination of two goal-reaching applications. As is shown through Figure 62, successive desired positions can be reached and the position error-signals subsequently converge to 0.

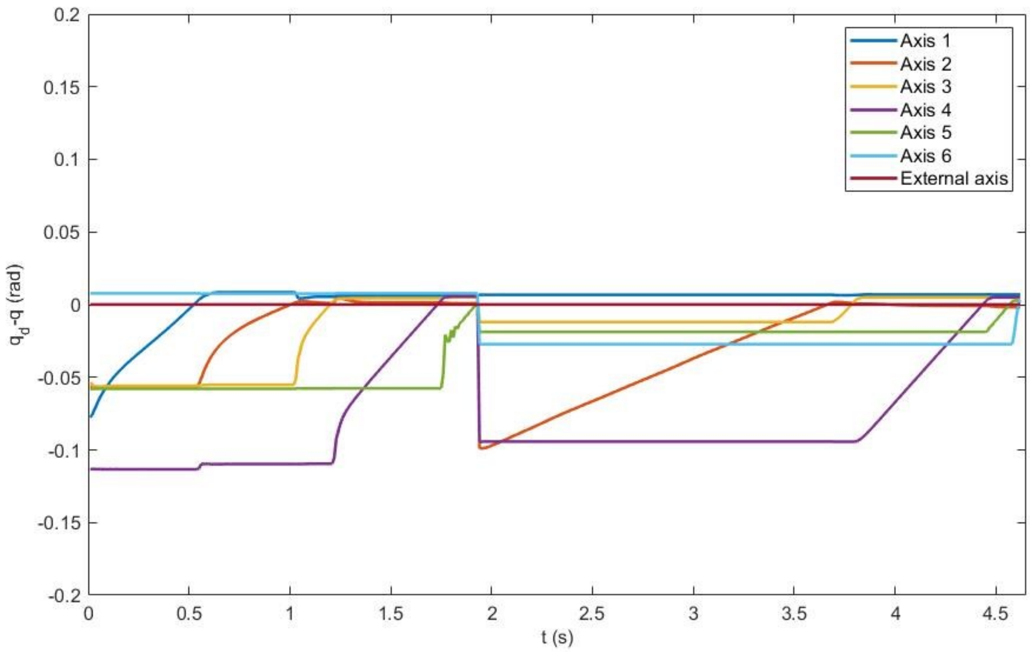



Figure 62: Position error.

The third use case is the following of a straight line. This is a uniaxial movement, i.e. only a single joint is moved. The angular position-error of the moved joint is illustrated in Figure 63. The error-signal repeatedly converges to 0 with a speed depending on the desired signals extracted from the .csv-file.

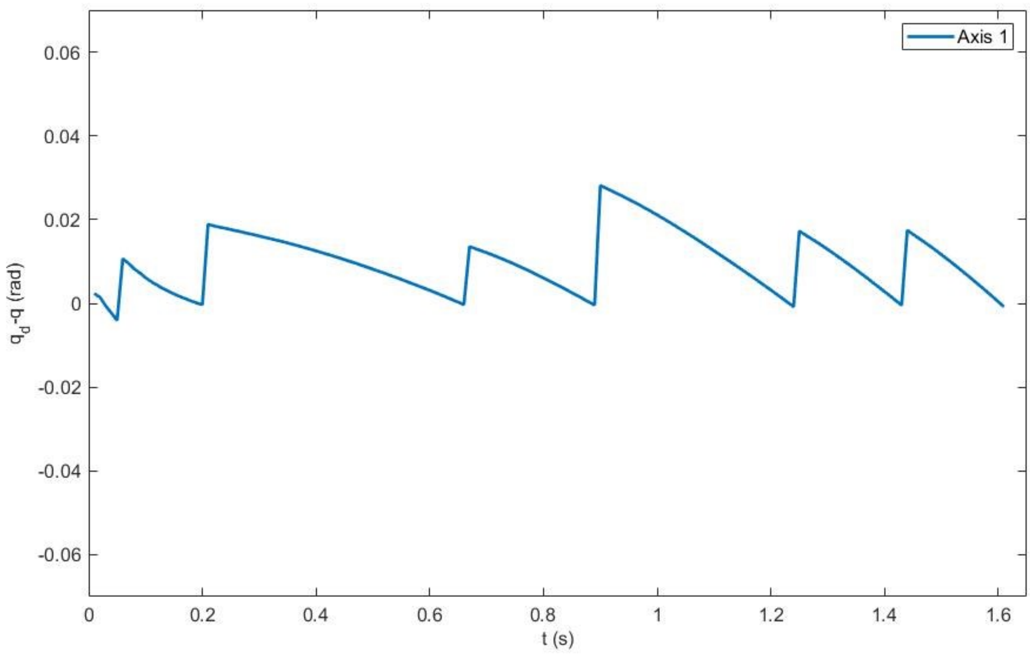



Figure 63: Position error.

The final experiment for the position control-part is a multiaxial trajectory tracking use case and combines the movement of several joints as illustrated in Figure 64. From the starting position (1st row left) a uniaxial movement, i.e. following of a straight line on the workpiece surface is performed (1st row right), then the end effector is twisted/inclined to continue the uniaxial line following (2nd row left) before the tool is lifted from the surface and reaches its goal position in free space (2nd row right). Figure 65 shows that the angular position-errors of all seven robot arm-joints successively converge to 0.

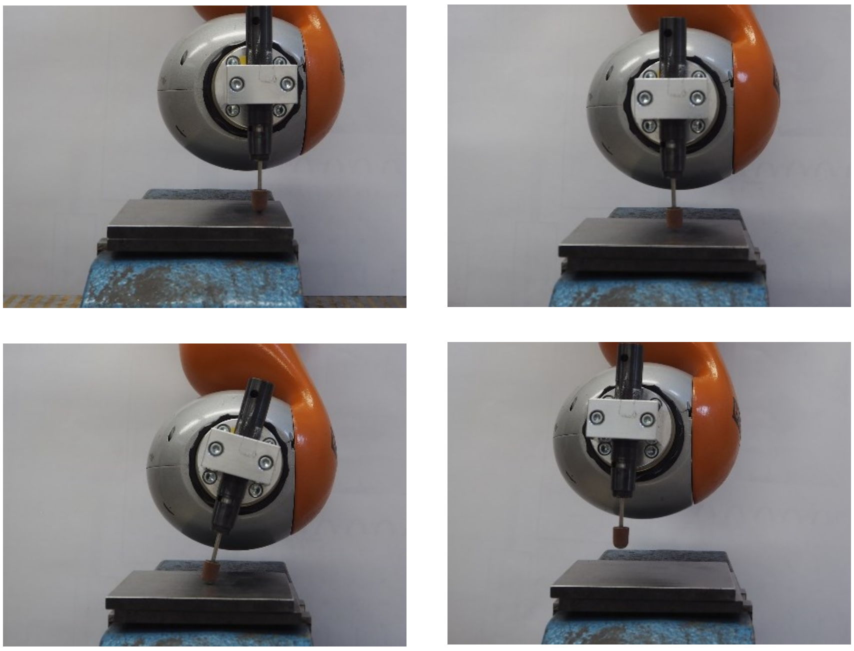


Figure 64: End-effector movement.

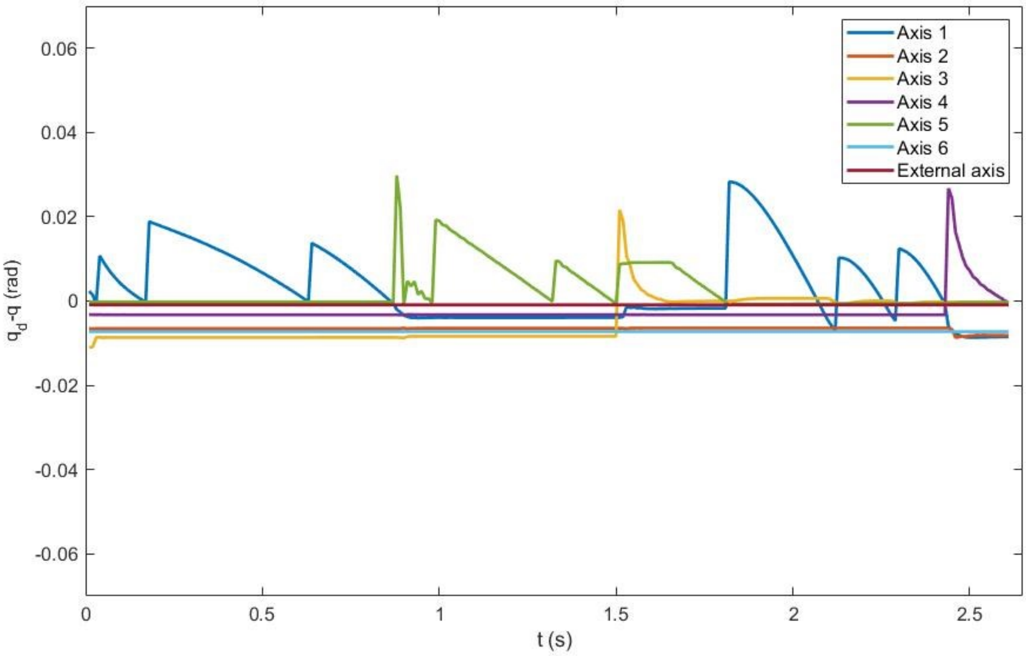



Figure 65: Position error.

Torque Control

To validate the torque controlling performance of the concept, it is chosen to compress bubble foil as illustrated in Figure 66 in a repetitive loop. The repetitive convergence to 0 of the error-values of the torque-signals is graphically illustrated in Figure 67.

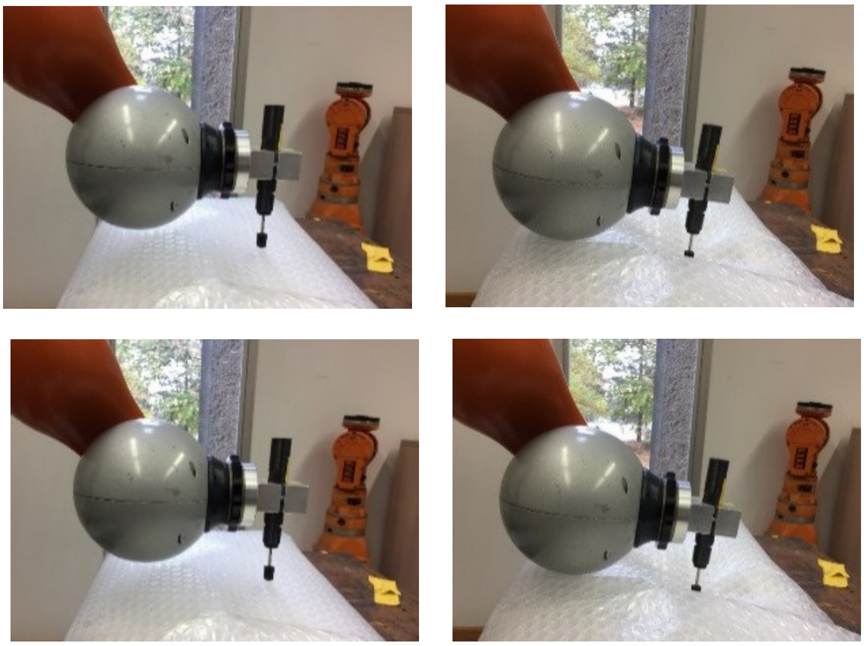


Figure 66: End-effector behaviour.

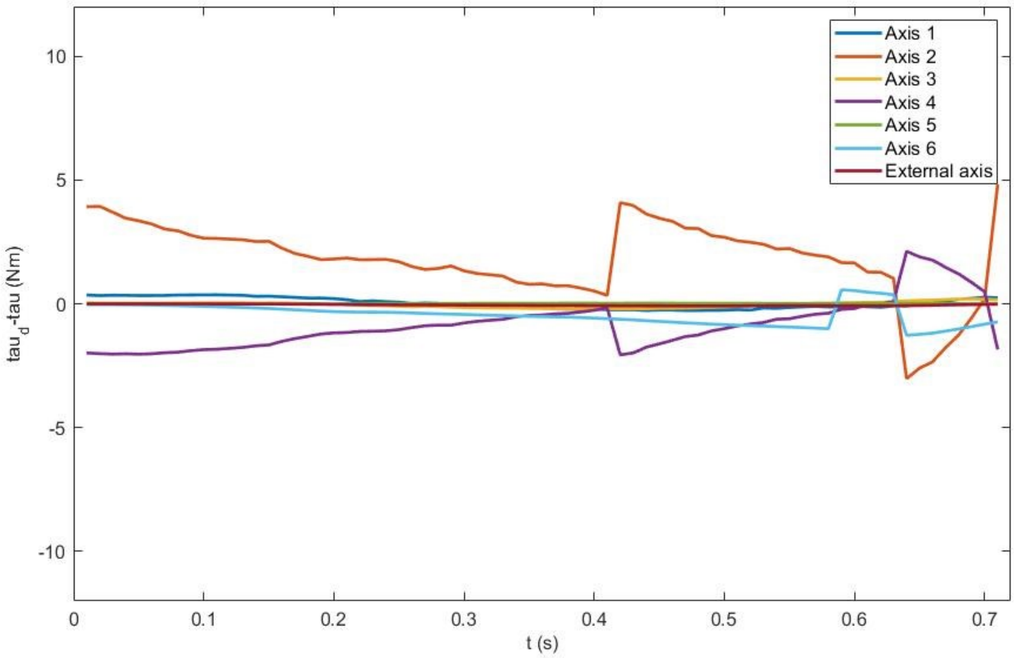



Figure 67: Torque error.

In a further experiment, the previous use case is repeated several times consecutively. Again the convergence to 0 of the difference between desired and current joint torques is shown (Figure 68).

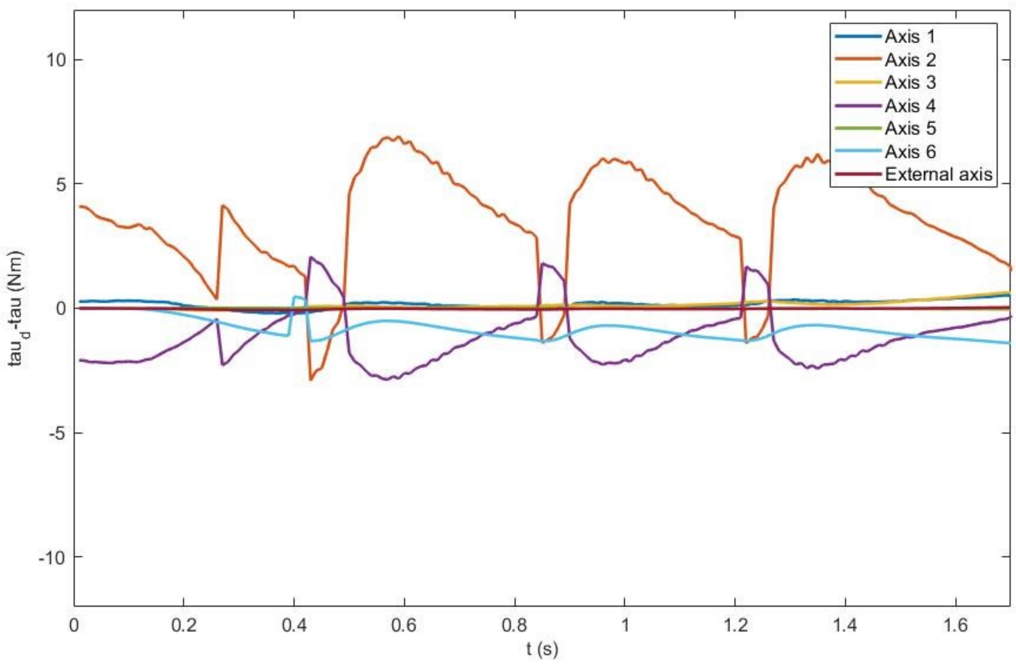



Figure 68: Torque error.

Combined Position and Torque Control

After the position- and torque-control performances have been validated individually, in a third phase the combined position- and torque-control is investigated.

First, the experiment from 6.2.3 is repeated. The motion sequence can be found in Figure 52. Figures 69 and 70 show the evolution of the error signals of angular position and torque, respectively. Both show convergence to 0.

A second experiment involves the actuation of more joints. Figure 71 depicts the phases of the process: from free space (1st row left), the end-effector is brought into contact with the surface (1st row right) and follows a straight line (2nd row left). The end-effector is lifted off the surface (2nd row right), moved before it tips the curved part of the workpiece (3rd row left). The end-effector is again lifted off the surface and moved in free space (3rd row right).

Figures 72 and 73 illustrate the convergence to 0 of the position- and torque-errors, respectively.

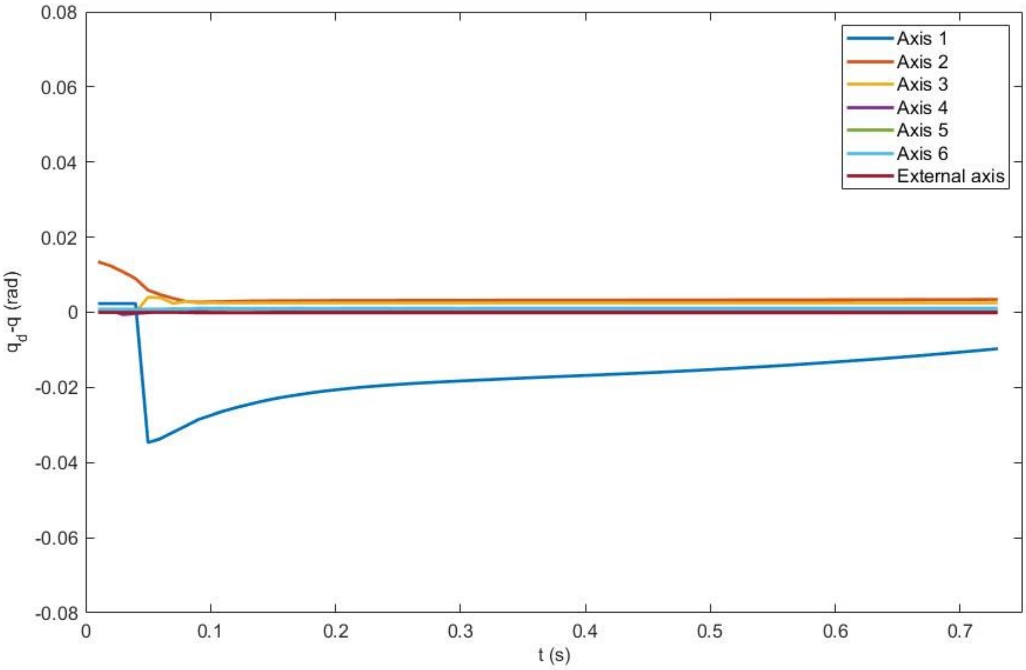



Figure 69: Position error.

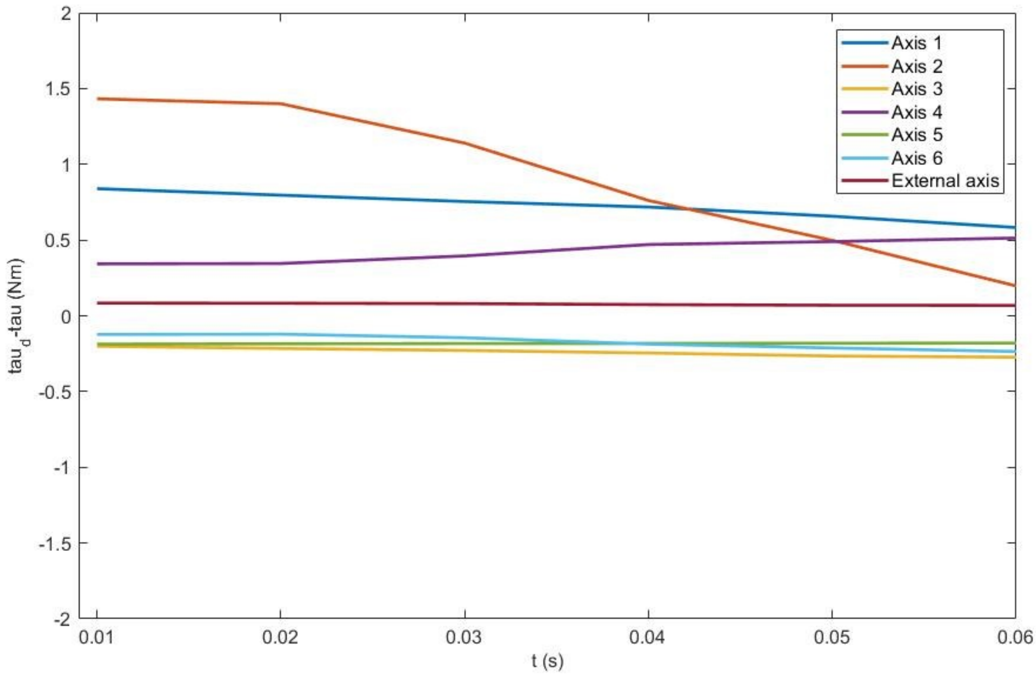



Figure 70: Torque error.

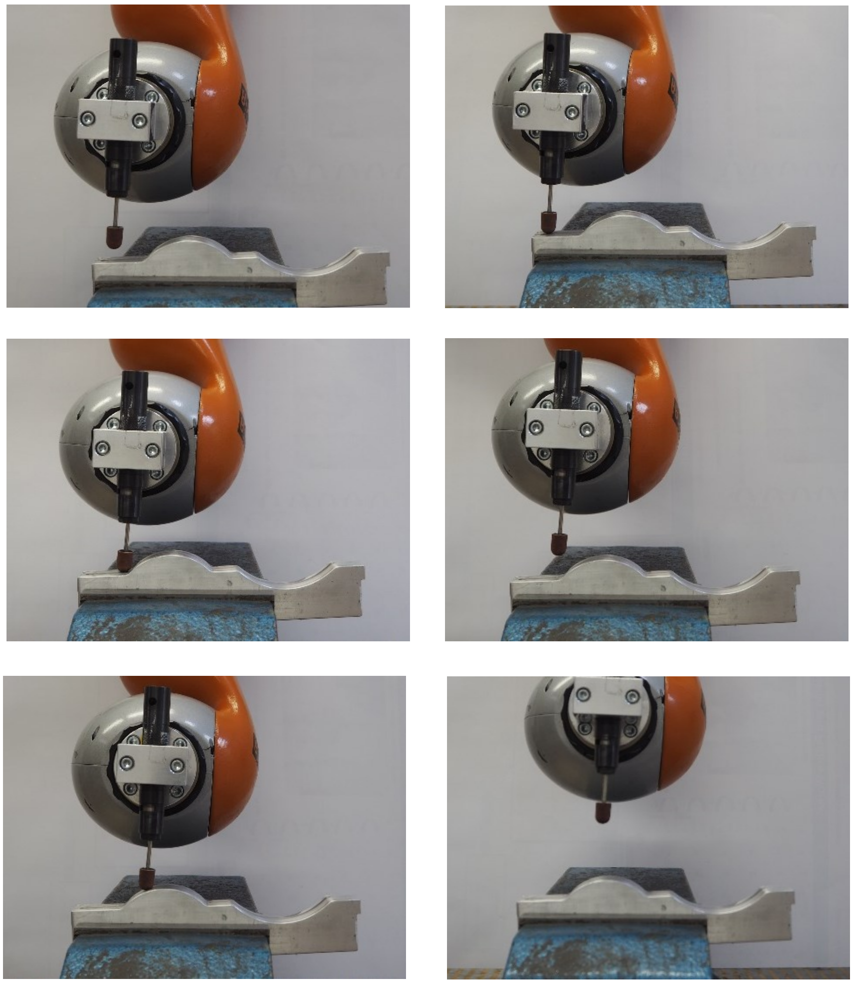


Figure 71: End-effector movement and behaviour.

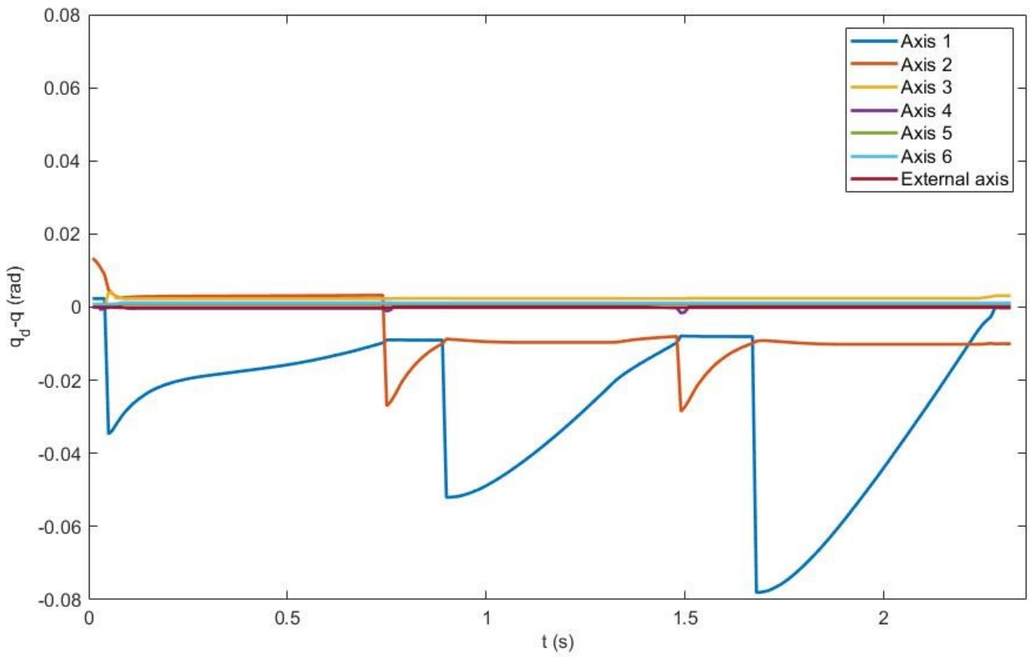



Figure 72: Position error.

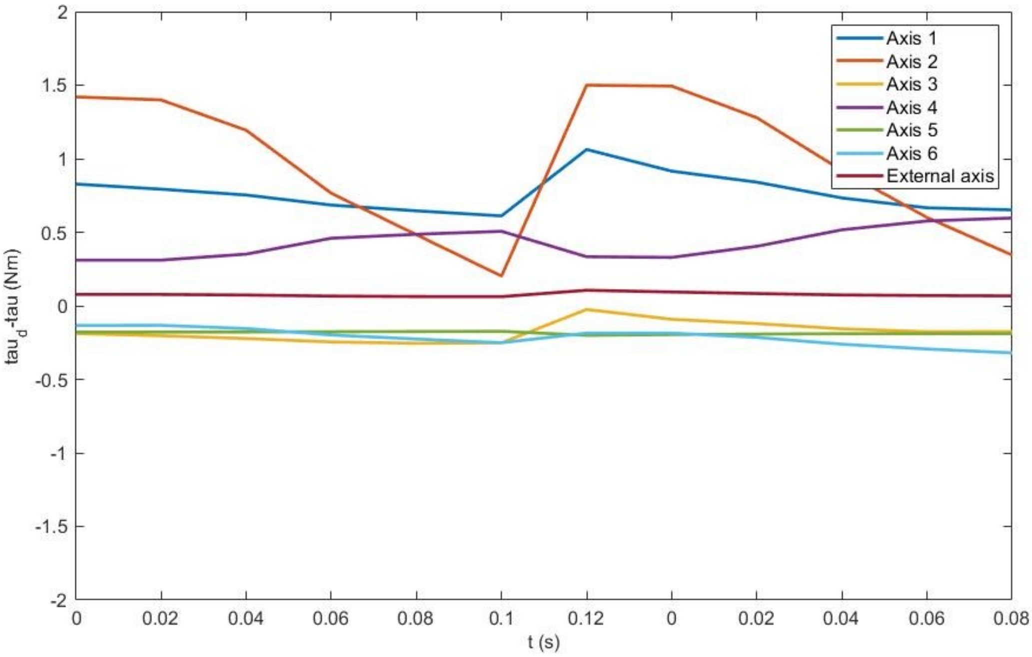



Figure 73: Torque error.

### 6.3.4 Analysis and Summary

In 6.3, only the most essential elements of the controller suggested in 6.2 are retained and combined into a new control concept. This controller is composed of two model-free controller-halves addressing respectively the position or torque control for a robot manipulator performing a freeform trajectory tracking application with the application of forces at specified positions. Both controller-halves combine conventional PID-control with bio-inspired learning. The latter is based on reinforcement learning and makes use of an actuator-inhibitor-system and a reward-like incentive. The aim is to expand the system's adaptability characteristics in case of dynamic environments through learning. The hypothesis is that the suggested concept outperforms simple PID-control and shows similar performance to the concept developed in 6.2 in constrained trajectory tracking despite the simplification.

6.3.1 is dedicated to the theoretical development of the model-free parallel control concept.

The twofold validation process starts with a numerical simulation (6.3.2). Compliant to the requirements set in 6.1.1, inputs for the simulated application are .csv-files, i.e. lists of successive desired joint positions and torques, respectively. The correspondence between the black solid lines and the blue dotted lines verifies the position and torque tracking. Additionally, it is shown that the suggested concept outperforms a parallel controller composed of two PID-controllers.

The second step of the validation process implies the implementation of the suggested control concept on the KUKA LWR4+-robotic manipulator introduced in 4.2. This validation step includes three sub steps. First, position control is validated individually based on 4 different experiments ranging from goal reaching over uniaxial line following to multiaxial movements including switching between contact and free space situations. The convergence to 0 of the position error-signals proves the position control performance. Second, torque control is tested individually. The differences between current joint torque and specified torque converge to 0 over time for all joints. After the individual performances of position and torque control of the suggested concept were analysed, their combination is investigated. As far as the input of desired torque signals is concerned, it was opted for the application of uniaxial torques at specified positions, similar to manufacturing forces. Finally, combined position and torque control is validated experimentally. As for the simulation, inputs were .csv-files, i.e. lists of successive desired joint positions and torques. The convergence to 0 of all error-signals is graphically presented. This proves that the desired values for positions and torques are reached. The parallel control concept is performing as expected and in compliance with the requirements of minimized mean position and torque errors established in 6.1.1.

Finally, it is shown that the simplified model-free version of the concept suggested and validated in 6.2 is able to address the combined position and torque control problem of constrained trajectory tracking by robotic manipulators. On the one hand, the suggested control concept outperforms simple PID-controllers. On the other hand, it complies with the desire for the simplest possible controller structure. Compared to the concept developed in 6.2, the simplified concept is model-free, i.e. neither model parameters nor NNs to estimate those are required. The performance qua absolute mean position error <0.3 rad and torque error <0.1 Nm is achieved. Further, the concept is in compliance with the requirement of human mimicry.

## 6.4 Discussion and Summary

This sixth chapter extended the position control concept of chapter 5 with force/torque control, i.e. the application of a desired torque is added to the previously analysed path following application. Contact-rich tasks cannot be addressed by pure position control as already slight deviations from the desired trajectory can lead to errors in the desired forces and torques (Abu-Dakka, 2015) (Kronander, 2014). This explains the interest in combined position and torque control (Phillips, 2016). The requirements pointed out in 6.1.1 involve the diminution of the differences between actual and desired angular position as well as joint torque. The focus is put on the minimization of the absolute mean error signals. Additional requirements are related to facilitated implementation in diverse settings. Among others, the requirement for the control concept to solely rely on robot-internal sensors is meant to first avoid dependency on external elements, i.a. sensors and to second facilitate the future implementation in industries susceptible to suffer from dusty environments which do not allow the use of most sensors. The suggested algorithms (6.2 and 6.3) follow each the design and validation processes as presented in Figure 46.

The theoretical development sets the analytical basis for the suggested control concepts. A twofold validation process follows. The first validations are performed in the Matlab/Simulink-environment on a RR robot arm (Figure 11). In a second step, the algorithms are validated through experiments on a setup including a KUKA LWR 4+-robot arm as introduced in 4.2.

Relevant experiments are path following use cases switching between free-space and in-contact positions and with applications of specific torques. Desired positions and torques are entered in form of tables with successive desired signals, as .csv-files to make sure the concepts are compatible with Programming by Demonstration. .csv-files are the common output of kinaesthetic teaching. The latter facilitates the transfer of human expertise to the robotic manipulator compared to e.g. tele-operation (Kormushev, 2011). Comparisons with conventional controllers, i.a. PID-controllers as described in 2.1.2 show the outperformance of the newly suggested algorithms. These controllers are chosen because they are, despite their simplicity still widely used in industrial applications and represent conventional control strategies. Linking back to the requirements introduced in 6.1.1, the improvements qua maximal and mainly mean error signals are proven. Absolute mean position errors are below the defined threshold value of 0.3 rad. As far as the absolute mean torque errors are concerned, the requirements of <0.1 Nm are met. Compared to conventional PID-control, the results are reduced by a factor superior to 10.

The hypothesis that combining a robust controller with a bio-inspired learning element improves the system-performance has been confirmed for position control in chapter 5. At this point, it was postulated that the suggested concept is extendable for combined position and torque control.

In 6.2, the concept developed in 5.4 was adopted for parallel position and torque control. Invariance to unknown robot models is transferred. 6.3 centers the attention on facilitated implementation by finding a balance between simple, but unsatisfactorily performing control (2.1.2) and highly performing, but complex control (6.2). This is achieved through focusing on the most essential elements of the complex concept only, i.e. combining conventional model-free PID-control with a learning element. The preference went out to model-free control algorithms in order to avoid any dependency on robot model knowledge. PID-control was chosen because of it being simple, well-known and field-tested. A reinforcement learning extension was added to improve the performance and adaptability of the conventional controller. Rather than presenting a true-to-life computational model of the human learning behaviour, the aim of the reinforcement learning-extension is to improve conventional PID-control.

This concludes the chapter on combined position and torque control and consequently also the part on the developments performed in the presented work. The performance and fulfilment of requirements of the suggested control concepts were validated through simulation and experiments. Finally, it was shown that with the combination of conventional control and learning element it is possible to address constrained trajectory tracking problems, i.e. path following with application of specified joint torques in adverse conditions like switching constraints or uncertainties. The improvement with respect to basic but still widely used controllers was proven.

# 7 Discussion and Future Work

The aims of this chapter are to

- point out the interdisciplinary aspects of the research,
- critically discuss the choices made in the presented work with respect to alternative avenues,
- position the work with respect to related literature and present it as a continuum of earlier research,
- and give recommendations for future research.

The presented research uses concepts and knowledge from different disciplines, in more or less direct ways. A non-exhaustive list is presented in this paragraph. While the starting point or use case was found in manufacturing engineering, the development of the control algorithms is built on theories and concepts from control engineering. Both disciplines intervene in the abstraction of the industrial challenge, in the formulation of the research questions and in the identification of requirements and objectives. The suggested algorithms combine conventional controllers from control theory with bio-inspired elements. The learning elements and approximation through Neural Networks involve concepts from computer science and more specifically from Artificial Intelligence. The inspiration for these algorithms originates from research in neurosciences. From psychology- and biology-research, abstracted models are derived which are then used for bio-inspired engineering in control algorithms. These models as well as the theoretical developments of the algorithms and the stability analyses apply mathematics. In addition, the used models of robot and environment are based on principles from mechanics and dynamics. For as far as the experimental performance evaluation is concerned, the robot hardware and sensors require electronics and mechatronics know-how. The communication between PC and robot as well as the implementation of the controller rely on informatics and programming knowledge.

One of the aims of this thesis was the exploration of a range of promising concepts for addressing trajectory tracking control. Different concepts have been investigated throughout the work and the choices made during the development of the control concepts have been justified to the best of the knowledge. However, not all possible directions of research were included. Therefore some of the most promising avenues which were deliberately left out of the main part of this work are discussed in the following paragraphs.

In a thesis on robotics touching upon learning and artificial intelligence, it is impossible to get around deep neural networks and deep reinforcement learning (LeCun, 2015). A central element of deep learning is the increased number of layers between input and output layer (Blanchot, 2019). At layer n the input for layer n+1 is computed as output. Next to this feedforward process, a backpropagation towards the input layer allows the weights to be updated. Deep learning is a prominent choice for addressing high-dimensional data as it allows to automatically find the low-dimensional representations (Arulkumaran, 2017). Distributed representation is one of the key features in deep learning: each input can be represented by many features and each of the latter can represent many inputs. Recently, deep learning has been a hot topic and has been applied to a vast range of fields and use cases (Goodfellow, 2016). A significant number of recent research works focuses on deep learning algorithms (Jonschkowski, 2017) including deep Neural Networks (Bezak, 2014) (Byravan, 2017) and deep reinforcement learning (Arulkumaran, 2017) (Li, 2018). Deep neural networks exhibit distributed structures, multiple layers and prominently use probabilistic relationships. Deep reinforcement learning applies deep neural networks for estimating its components, i.e. value function, policy or model. Among others, deep learning and more specifically deep neural networks were applied to classification or recognition tasks (Liu, 2018) and to multi sensor-information fusion (Bohez, 2017). Deep reinforcement learning has been applied to areas ranging from natural language processing over computer vision to robotic control (Goodfellow, 2016). Qua data-efficiency, the main downsides of deep learning algorithms are the training period and required considerable amount of training data samples (Arulkumaran, 2017) (Sünderhauf, 2018). Related drawbacks include high computation time and overfitting. Further, despite significant advancements in algorithms and computational power, no general intelligence has been achieved so far. Artificial intelligence remains limited to a single task it has been programmed for (Blanchot, 2019).

Regarding the specific application and the related requirements in this work, scaling up to a deep NN or deep reinforcement learning might benefit the estimation capacity of the unknown robot parameters and the tracking performance of the system. This assumption is based on the improved convergence and performance in learning dynamics and approximating functions as shown by e.g. (Silver, 2017), (Yamaguchi, 2016) and (Gu, 2016), respectively. For the specific application considered in this work, however, the suggested simple perceptron model and reinforcement learning concept show satisfactory performance. They fulfil the established requirements next to presenting the advantages of simplicity, low computational costs and minimized training phases and data. The expected improvements of upscaling to deep learning are not expected to be worth accepting the additional effort and downsides. A clear, data-efficient concept is preferable. However, scaling up to deep learning for different applications is a recommendation for future work (LeCun, 2015).

The presented work assumes repetitive workpiece-locations and –conditions, i.e. the control concept always starts from the same position with respect to the goal-surface. Before performing a surface finishing process, human operators analyse the workpiece-conditions and react accordingly, i.e. they adapt the contact-situation of the tool with respect to the surface. Its capability to react to different situations with specific control rules, makes the human operator highly valuable for processes with varying parameters (Kiguchi, 2002). This first step, i.e. adaptability of the human behaviour in altered contact-situations or problematic cases is ignored in the suggested control concepts. Their performance is optimized for the general process thanks to adaptive position- and force-/torque-control algorithms, but they are unable to address specific problematic cases as often encountered in practice because of variations in workpiece-dimensions or –positioning. For specific, problematic cases, additional elements are necessary (Weinert, 2007). Figure 74 illustrates potential deviations from the expected situations: workpiece too close, end effector too far or interference of an obstacle.

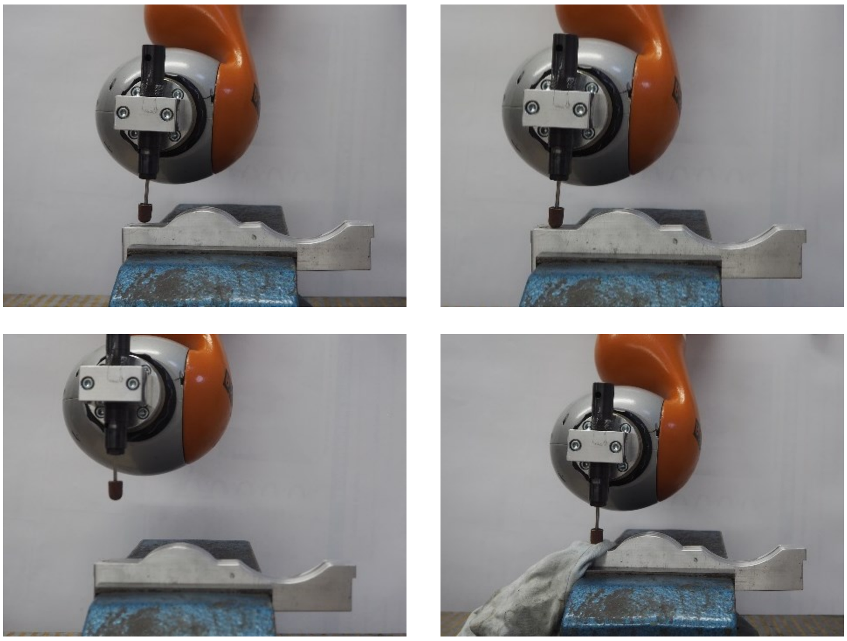


Figure 74: Expected workpiece location and potential deviations.

In a first phase, humans apply rules of thumb in the form of 'if a then do b'. From previous experience, they deduce a rule base containing generalized guidelines about what actions to take in what circumstances, i.e. if-then statements (Dayan, 2014) (Franchi, 2016) (Passino, 2005). The desire to mimic this human behaviour culminated in rule-based systems, one of the most straightforward forms of artificial intelligence. Knowledge is encoded into these systems as a set of rules that specify how to act in different situations. These rules are declared as an array of if-then rules: 'if a then b' or 'a => b'. They can be described as the linguistic formulation of the human's approach to achieve a goal. A rule-based system is designed to mimic the behaviour of a human expert when facing a specific challenge. The performance of a rule-based system is therefore expected to be similar to that of a human expert in the considered area and when exposed to the same data (Grosnan, 2011) (Noroozi, 2009). A priori programmed relational condition-action statements have been implemented i.a. in behaviour-based robotics (Matarić, 1998) and probabilistic contexts (Kapotoglu, 2015). Fuzzy logic is a prominent method to express quantitative aspects of a human expert's knowledge and reasoning process with a set of linguistic rules (Mishra, 2010). Fuzzy rules have been applied for acquiring knowledge bases (Huang, 2013) (Li, 2015), approximating unknown plant models in combination with Artificial Neural Networks (He, 2018), online task sequencing in robotic assembly in combination with Petri nets (Cao, 1994), extending a sliding mode control algorithm (Ghalyan, 2016) (Jasim, 2015) (Roopaei, 2009) (Sun, 1999), clustering in combination with particle swarm algorithms (Szabo, 2012), modelling trajectories (Skoglund, 2010) or machining processes (Mohd Adnan, 2015). Interaction and behavioural rules in social animals and insects have inspired algorithms for multi agent behaviour and clustering

(Passino, 2005). (Kao, 2008) addressed a clustering problem with an algorithm inspired on ant colonies. The elements evolve according to a limited number of behavioural rules in the form of if-then statements. Other algorithms were inspired on the behavioural rules of fish swarms (Huang, 2013), bird flocks (Cui, 2006) (Cui, 2006b) (Forestiero, 2013) (Reynolds, 1987) and ant colonies (Elkamel, 2015) (Vaijayanthi, 2011).

Next to the dependence on expert systems and the restriction to if-then statements, the main drawback of rule-based systems is the limitation of the number of rules for computational and complexity reasons (Grosnan, 2011) (Kiguchi, 2002) (Li, 2010) (Li, 2015) (Roopaei, 2009) (Sun, 1999). However, as a rule-based system allows the encoding of expert knowledge in a narrow area of the specified problem and can be implemented in a hierarchical way to survey a previously developed control system (Passino, 2005), an extension of the suggested control concept (6.3) is well conceivable. Human operators tackle specific cases of altered workpiece-conditions by observing the situation (if a) and reacting appropriately (then b). This step is performed before the rest of the control approach, as an outer supervising loop. The challenge consists in adding this lacking element into the control concept of the automated process. The specific goal of this future work is to add a control element able to deal with particular problematic situations, e.g. uncertainties in workpiece position and dimension. The requirement for the considered manufacturing task is 1) to keep contact between workpiece and tool, i.e. not to lift off and apply the manufacturing force only once contact has been established 2) not to damage the workpiece through penetration of the tool, i.e. to back up once a force threshold has been surpassed. The recurrent theme identified during the observation of the human operator is the following approach: first observe then react. Before performing any action, humans analyse and categorise the situation at hand and take a decision about the optimal approach or reaction based on their experience. This behaviour can be represented as if-then-rules: if situation a is identified then action b is chosen as optimal reaction. This becomes particularly valuable in problematic cases which deviate from the general or pre-planned case. A superposed rule-based control element should be able to address the identified problematic cases before the other controller-elements come into effect. This new control element supervises the process controlled by the algorithm (6.3) in a hierarchical manner. This outer control loop is derived from the observations of human reasoning and implements human experience and knowledge in the form of rules as shown in Figure 75 and Pseudo-code below (Klecker, 2019).

| Code | Description |
|---|---|
| while $F_{contact}$ > threshold | *in case of risk of penetration* |
| { | |
| position = position – increment | *lift the tool* |
| } | |
| if a manufacturing force is desired | *condition* |
| { | |
| while $F_{contact}$ < threshold | *in case of no contact* |
| { | |
| position = position + increment | *approach the surface* |
| } | |
| } | |

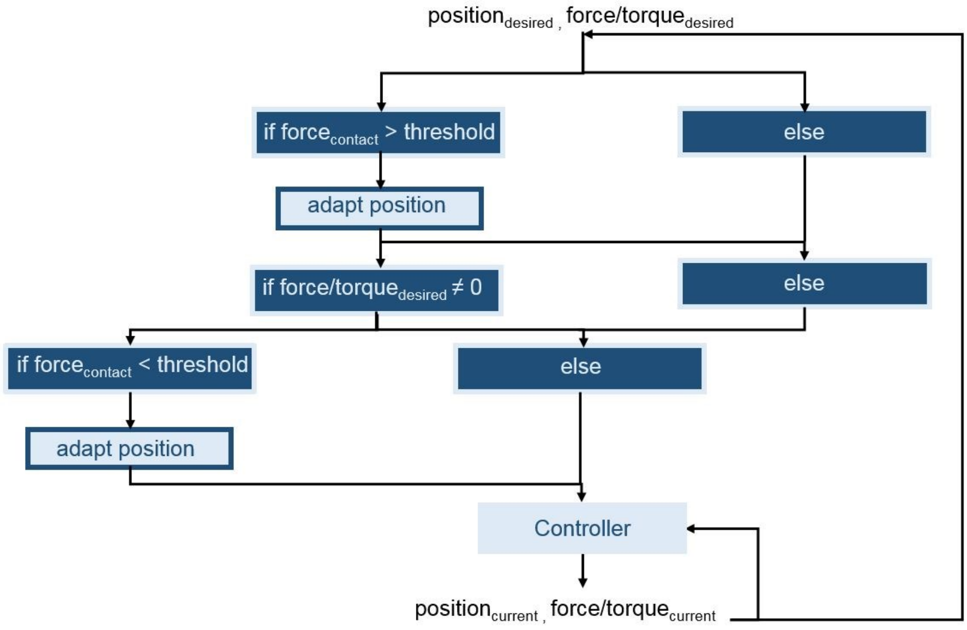



Figure 75: Block diagram of the hierarchical integration of a rule-based extension in the control concept.

Implementation and further investigation of this suggested controller-extension are possible topics for future work.

Another point for discussion concerns the considered application, i.e. an automated surface finishing process. A straightforward way to provide the robot system with the necessary skills is through Programming by Demonstration or kinaesthetic teaching. In this method the human operator teaches the robot which paths to follow and where to apply which forces by showing and recording the process. This technique passes the operator's know-how on to the robot thereby taking advantage of his expertise. The focus of this approach lies on transferring the human knowledge to the automated system. The related downsides are dependence on the human skills and limitation to the specific taught process. An alternative approach to design an automated process able to achieve the desired surface finishing results focusses on the manufacturing process. Through modelling the surface finishing process, the required manufacturing forces can be computed and used as inputs for the torque controller. Models of the manufacturing process including its specific stability- and quality-requirements can be found in earlier works by i.a. (Benardos, 2003) (Brinksmeier, 2006) (Inasaki, 2001) (Jiang, 2013) (Klocke, 2005) (Kurfess, 1988) (Stepien, 2009). The advantage of this method is the focus on the manufacturing process and its results. This leads, however, to a dependence on a time-consuming modelling process. Further, this technique is strictly restricted to the modelled manufacturing process and does not take into account human expertise. As the vision of the presented work includes human mimicking in intuitive automation, PbD or kinaesthetic teaching was chosen.

In this work, the compliance with records from Programming by Demonstration was shown. The common output of kinaesthetic teaching, .csv-files were used as input for the suggested control concepts. Further exploitation of Programming by Demonstration-features like optimization on the task-level strategy (Chen, 2005), additional involvement of the human (Aleotti, 2006) (Calinon, 2007) (Castelli, 2007) (Rozo, 2016) or the use of movement primitives for the generalization of skills (Deniša, 2016) (Kramberger, 2017) (Montebelli, 2015) (Zhang, 2018) can be found in the dedicated literature and present promising avenues for extension and future work.

Contiguous to the technical aspects, an ethical analysis should be considered when intending the implementation of robots or an automated process in an industrial setting. Technological progress related to robotics and artificial intelligence is a fact and the impact on the manufacturing industry and its labour cannot be denied. Contrary to the panic fomented by mass media evoking the new and imminent threat of science fiction-like scenarios of humanoid robots taking over the lead and oppressing human workers, automation is not a new trend. The replacement of human workforce is a trend which has been ongoing over the past century (Wyatt, 2006). Despite technology's contribution to the improvement of jobs in terms of ergonomics and safety, there exists a pluralism in society. While some argue for the unrestricted exploitation of technological opportunities, others are in favour of banning all automation in industry and maintaining a status-quo of the current situation. As neither of these extreme scenarios is likely to happen, experts and society alike should be prepared for the introduction of robots in manufacturing engineering and find a consensus among all stakeholders with their diverging opinions leading to a global framework for the temporal and local coexistence of human and robotic workforce in an industrial environment. Robot ethics and machine ethics are the two main branches of ethics interested in similar discussions. On the one hand, the interdisciplinary field of robot ethics is concerned with the impacts of robots on society, i.e. it deals with ethical questions related to the emergence of the robotic industry (IEEE, 2018) (Lin, 2011) (Scheutz, 2013) (Veruggio, 2008). Machine ethics on the other hand, is concerned with the ethical behaviour of machines or robots (Anderson, 2011) (Brundage, 2014) (Tonkens, 2009). Rather than investigating the ethical consequences of robotic assistants for society and targeting the design process by human developers, machine ethics investigates the ethical behaviour of the robots and targets the ethical guidelines which are implemented (Bryson, 2017) (Bryson, 2017b) (Deng, 2015) (Wallach, 2010). Although this topic should not be neglected, a complete ethical analysis is out of the scope of this thesis and therefore left for future work by experts in the relevant fields.

The seven chapters of this thesis conclude with these ideas for future research projects.

## Summary

In the presented work, control algorithms for pure position and combined position and force/torque control for freeform trajectory tracking were developed in a bottom-up approach.

Conventional controllers satisfactorily address basic path following applications in close to ideal circumstances. Due to their simple design and straightforward implementation, they are the most used control concepts in industry. But their stability and performance are deteriorated in case of adverse conditions like switching constraints or uncertainties. Therefore, to address a wider range of trajectory tracking applications, a more complex control concept was developed. A combination of robust, basic control and a reinforcement learning based control-extension merges the constituents' respective advantages while overcoming their drawbacks.

The aim of position control was to keep absolute mean position error below 0.3 rad for all robot joints in case of freeform shapes, switching constraints and unknown robot dynamics. In 5.2, Sliding Mode Control was extended with BELBIC. The concept modelling the mammalian emotional learning behaviour via the interplay of amygdala and orbitofrontal cortex, had been introduced by (Lucas, 2004) in the early 21st century. For a priori known robot models, position errors far below the set threshold (0.3 rad) were achieved. The suggested controller outperformed model-free and model-based conventional controllers. In 5.3, potentially unknown robot parameters were estimated as a single robot function using a Radial Basis Function Neural Network. The desired performance was achieved in the simulations despite the lack of robot dynamics knowledge. In 5.4, a RBF-NN with online update laws for the parameters was implemented. The requirements qua performance were met in simulation and experiment, independent of unknown robot models and NN parameters initialization.

Combined position and force control improves the trajectory tracking performance and allows the application of desired torques at specified positions. In 6.2, the concept developed in 5.4 was adopted for parallel position and torque control. Absolute mean position errors were kept below 0.3 rad and torque errors below 0.1 Nm. In 6.3, a balance was found between simple, but unsatisfactorily performing control (2.1.2) and highly performing, but complex control (6.2). Focusing on the most essential elements of the complex concept, resulted in a model-free parallel concept combining PID-control with reinforcement learning-based features. Simulations and experiments validated both stability and performance. Absolute mean position error was kept below 0.3 rad and torque error below 0.1 Nm as specified in the requirements.

Finally, the synergy of classical control concepts with bio-inspired learning elements was able to address the trajectory tracking control problem by combining their characteristics: robustness and adaptability. The objectives in terms of performance were achieved in simulation and experiment (mean absolute position and torque errors < 0.3 rad and 0.1 Nm for all joints). The objective of independence of robot models was reached by approximating unknown parameters as a single nonlinear robot function or by model-free algorithms. The same requirements qua position and torque errors were met despite the lack of robot model knowledge. Human expertise was incorporated through the bottom-up approach: first position control and only then position control combined with the application of forces. Controller-inputs generated from kinaesthetic teaching took advantage of the know-how of human operators.

## References

Abu-Dakka F. J., Nemee B., Jorgensen J. A., Savarimuthu T. R., Krüger N., Ude, A., 2015, ‘Adaptation of manipulation skills in physical contact with the environment to reference force profiles’, Autonomous Robots, 39, 2, pp. 199-217.

Adar N., Kozan R., 2016, ‘Comparison between Real Time PID and 2-DOF PID Controller for 6-DOF Robot Arm.’, Acta Physica Polonica A, 130, 1, pp. 269-271.

Akgun B., Thomaz A., 2016, ‘Simultaneously learning actions and goals from demonstration’, Autonomous Robots, 40, 2, pp. 211-227.

Albu-Schäffer A., Haddadin S., Ott C., Stemmer A., Wimböck T., Hirzinger G., 2007, ‘The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments’, Industrial Robot, 34, 5, pp. 376-385.

Aleotti J., Caselli S., 2006, ‘Robust trajectory learning and approximation for robot programming by demonstration’, Robotics and Autonomous Systems, 54, pp. 409-413.

Alontseva D. L., Ghassemieh E., Krasavin A. L., Shadrin G. K., Kussaiyn-Murat A. T., Kadyroldina A. T., 2019, ‘Development of Control System for Robotic Surface Tracking’, Proceedings – ecmmm.

Anderson M., Anderson S. L., 2011, ‘Machine Ethics’, Cambridge: Cambridge University Press.

Argall B. D., Chernova S., Veloso M., Browning B., 2009, ‘A survey of robot learning from demonstration’, Robotics and Autonomous Systems, 57, pp. 469-483.

Arimoto S., 1994, ‘State-of-the-art and future research directions of robot control’, Preprints of the Fourth IFAC Symposium on Robot Control, 27, 14, pp. 3-14.

Arimoto S., 1994b, ‘A Class of Quasi-Natural Potentials and Hyper-Stable PID Servo-Loops for Nonlinear Robotic Systems’, DOI: 10.9746/sicetr1965.30.1005.

Arimoto S., 1996, ‘Control Theory of Nonlinear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach’, New York: Oxford University Press.

Arulkumaran K., Deisenroth M. P., Brundage M., Bharath A. A., 2017, ‘Deep Reinforcement Learning – A brief survey‘, IEEE Signal Processing Magazine, 34, 6, pp. 26-38.

Åström K. J., Wittenmark B., 2008, ‘Adaptive Control, Second Edition’, Mineola: Dover.

Baldassarre G., Mannella F., Fiore V. G., Redgrave P., Gurney K., Mirolli M., 2013, ‘Intrinsically motivated action-outcome learning and goal-based action recall: A system-level bio-constrained computational model’, Neural Networks, 41, pp. 168-187.

Balkenius C., Morén J., 2001, ‘Emotional Learning: a Computational Model of the Amygdala’, Cybernetics and Systems 32, 6, pp. 611-636.

Barto A. G., 2019, ‘Reinforcement Learning: Connections, Surprises, Challenges’, AI Magazine, Spring 2019, 40, 1, pp. 3-15.

Barve A., Nene M. J., 2013, ‘Survey of Flocking Algorithms in Multi-agent Systems’, IJCSI International Journal of Computer Science Issues, 10, 6, 2, pp. 110-117.

Bass E., Lee K. Y., 1994, ‘Robust Control of Nonlinear Systems Using Norm-Bounded Neural Networks’, IEEE World Congress on Computational Intelligence (Neural Networks part), pp. 2524-2529.

Benardos P. G., Vosniakos G., 2003, ‘Predicting surface roughness in machining: a review‘, International Journal of Machine Tools & Manufacture, 43, pp. 833-844.

Bertsekas D. P., Tsitsiklis J. N., 1996, ‘Neuro-Dynamic Programming‘, Belmont: Athena Scientific.

Bezak P., Bozek P., Nikitin Y., 2014, ‘Advanced Robotic Grasping System Using Deep Learning’, Procedia Engineering, 96, pp. 10-20.

Bischoff R., Kurth J., Schreiber G., Koeppe R., Albu-Schäffer A., Beyer A., Eiberger O., Haddadin S., Stemmer A., Grunwald G., Hirzinger G., 2010, ‘The KUKA-DLR Lightweight Robot arm-a new reference platform for robotics research and manufacturing’, ISR/Robotik, pp. 741-748.

Blanchot V., 2019, ‘Intelligence artificielle, machine learning, deep learning: quelles différences?’, Intelligence Artificielle, Siècle digital, https://siecledigital.fr/2019/01/30/differences-intelligence-artificielle-machine-learning-deep-learning/?sfns=st, visited 01.02.2019.

Bohez S., Verbelen T., De Coninck E., Vankeirsbilck B., Simoens P., Dhoedt B., 2017, ‘Sensor Fusion for Robot Control through Deep Reinforcement Learning’, arXiv: 1703.04550v1 [cs.RO].

Brinksmeier E., Aurich J. C., Govekar E., Heinzel C., Hoffmeister H.-W., Klocke F., Peters J., Rentsch R., Stephenson D. J., Uhlmann E., Weinert K., Wittmann M., 2006, ‘Advances in modeling and simulation of grinding processes‘, CIRP Annals-Manufacturing Technology, 55, 2, pp. 667-696.

Broomhead D. S., Lowe D., 1988, ‘Multivariable Functional Interpolation and Adaptive Networks’, Complex Systems, 2, pp. 321-355.

Brundage M., 2014, ‘Limitations and risks of machine ethics’, Journal of Experimental & Theoretical Artificial Intelligence, 26, 3, pp. 355-372.

Bryson J. J., Winfield A., 2017, ‘Standardizing Ethical Design for Artificial Intelligence and Autonomous Systems’, Computer, 50, 5, pp. 116-119.

Bryson J., 2017b, ‘AI Ethics: Artificial Intelligence, Robots, and Society’, ILIAS Distinguished Lecture Series 2017 – Final Report, Bouvry P., Bisdorff R., Schommer C., Sorger U., Theobald M., van der Torre L., University of Luxembourg, Luxembourg.

Byravan A., Fox D., 2017, ‘SE3-Nets: Learning Rigid Body Motion using Deep Neural Networks’, arXiv: 1606.02378v3 [cs.LG].

Calinon S., 2009, ‘Robot programming by demonstration: a probabilistic approach’, Lausanne: EPFL Press.

Calinon S., Billard A., 2007, ‘Active Teaching in Robot Programming by Demonstration’, Proceedings - IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).

Cao T., Sanderson A. C., 1994, ‘Task Decomposition and Analysis of Robotic Assembly Task Plans Using Petri Nets’, IEEE Transactions on Industrial Electronics, 41, 6, pp. 620-630.

Castelli V., Oblinger D., Bergman L., 2007, ‘Augmentation-Based Learning combining observations and user edits for Programming-by-Demonstration’, Knowledge-Based Systems, 20, pp. 575-591.

Cervantes I., Alvarez-Ramirez J., 2001, ‘On the PID tracking control of robot manipulators’, Systems & Control Letters 42, pp. 37-46.

Chen J. R., 2005, ‘Constructing Task-Level Assembly Strategies in Robot Programming by Demonstration’, The International Journal of Robotics Research, 24, 12, pp. 1073-1085.

Chen S., Billings S. A., Cowan C. F. N., Grant P. M., 1990, ‘Practical Identification of NARMAX Models Using Radial Basis Functions’, Research Report Acse Report 393, Department of Automatic Control and System Engineering, University of Sheffield.

Chen Y., Ma G., Lin S., Gao J., 2012, ´Adaptive Fuzzy Computed-Torque Control for Robot Manipulator with Uncertain Dynamics‘, International Journal of Advanced Robotic Systems, 9, DOI: 10.5772/54643.

Chiaverini S., Sciavicco L., 1993, ‘The Parallel Approach to Force/Position Control of Robotic Manipulators, IEEE Transactions on Robotics and Automation, 9, 4, pp. 361-373.

Craig J. J., 2005, ‘Introduction to Robotics Mechanics and Control’, Upper Saddle River: Pearson Prentice Hall.

Cui X., Potok T. E., 2006, ‘A Distributed Agent Implementation of Multiple Species Flocking Model for Document Partitioning Clustering’, CIA 2006, LNAI 4149, pp. 124-137.

Cui X., Gao J., Potok T. E., 2006b, ‘A flocking based algorithm for document clustering analysis’, Journal of Systems Architecture, 52, pp. 505-515.

Da Silva M. M., Brüls O., Swevers J., Desmet W., Van Brussel H., 2009, ´Computer-aided integrated design for machines with varying dynamics‘, Mechanism and Machine Theory, 44, pp. 1733-1745.

Daryabeigi E., Abjadi N. R., Arab Markadeh G. R., 2014, ‘Automatic speed control of an asymmetrical six-phase induction motor using emotional controller (BELBIC)‘, Journal of Intelligent & Fuzzy Systems, 26, pp. 1879-1892.

Dayan P., Berridge K. C., 2014, ‘Model-Based and Model-Free Pavlovian Reward Learning: Revaluation, Revision and Revelation’, Cognitive  Affective & Behavioral Neuroscience Journal, 14, 2, pp. 473-492.

Deisenroth M. P., Fox D., Rasmussen C. E., 2015, ‘Gaussian Processes for Data-Efficient Learning in Robotics and Control‘, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37, 2, pp. 408-423.

Deng B., 2015, ‘The Robot’s Dilemma-Working out how to build ethical robots is one of the thorniest challenges in artificial intelligence’, Nature, 523, pp. 24-26.

Deniša M., Gams A., Ude A., Petrič T., 2016, ‘Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization’, IEEE/ASME Transactions on Mechatronics, 21, 5, pp. 2581-2594.

Desborough L. D., Miller R. M., 2002, ‘Increasing customer value of industrial control performance monitoring—Honeywell’s experience’, Chemical Process Control – VI (Tuscon, Arizona, Jan. 2001), AIChE Symposium Series, 326, 98, USA.

Di Steffano J. J., Stubberud A. R., Williams I. J., 1967, ‘Feedback and Control Systems’, Schaum’s Outline Series, New York: McGraw-Hill Book Company.

Dieste J. A., Fernández A., Roba D., Gonzalvo B., Lucas P., 2013, ‘Automatic grinding and polishing using Spherical Robot’, Procedia Engineering, 63, pp. 938–946.

Domroes F., Krewet C., Kuhlenkotter B., 2013, ‘Application and Analysis of Force Control Strategies to Deburring and Grinding‘, Modern Mechanical Engineering, 3, pp. 11-18.

Duan Y., Andrychowicz M., Stadie B., Ho J., Schneider J., Sutskever I., Abbeel P., Zaremba W., 2017, ‘One-Shot Imitation Learning’, arXiv: 1703.07326v3 [cs.AI].

Elkamel A., Gzara M., 2015, ‘A bio-inspired hierarchical clustering algorithm with backtracking strategy’, Applied Intelligence, 42, pp. 174-194.

Englert P., Toussaint M., 2018, ‘Learning manipulation skills from a single demonstration’, The International Journal of Robotics Research, 37, 1, pp. 137-154.

Faber M., Bützler J., Schlick C. M., 2015, ‘Human-robot cooperation in future production systems: Analysis of requirements for designing an ergonomic work system‘, Procedia Manufacturing, 3, pp. 510-517.

Fazeli M., Sadigh M. J., 2012, ‘Adaptive Hybrid Position/Force Control for Grinding Applications’, Proceedings - IEEE International Conference on Technology in Automation, Control and Intelligent Systems, pp. 297-302.

Feng H., Su N., 2000, ‘Integrated tool path and feed rate optimization for the finishing machining of 3D plane surfaces’, International Journal of Machine Tools & Manufacture, 40, pp. 1557–1572.

Ferguene F., Achour N., Toumi R., 2009, ‘Neural network parallel force/position control of robot manipulators under environment and robot dynamics uncertainties’, Archives of Control Sciences, 19, 1, pp. 93-121.

Figueira L. B., Roque A. C., 2011, ‘Pattern Recognition Using a Recurrent Neural Network Inspired on the Olfactory Bulb’, IWINAC 2011, Part II, LNCS 6687, pp. 275-285.

Fitts P. M., 1954, ‘The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement’, Journal of Experimental Psychology, 47, 6, pp. 381-391.

Forestiero A., Pizzuti C., 2013, ‘A single pass algorithm for clustering evolving data streams based on swarm intelligence’, Data Mining and Knowledge Discovery, 26, pp. 1-26.

Franchi A. M., Mutti F., Gini G., 2016, ‘From learning to new goal generation in a bioinspired robotic setup’, Advanced Robotics, 30, 11, 12, pp. 795-805.

Frank M., Leitner J., Stollenga M., Förster A., Schmidhuber J., 2014, ‘Curiosity driven reinforcement learning for motion planning on humanoids‘, Frontiers in Neurorobotics, 7, pp. 1-15.

Fu J., Levine S., Abbeel P., 2016, ‘One-Shot Learning of Manipulation Skills with Online Dynamics Adaptation and Neural Network Priors’, Proceedings - IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 4019-4026.

Gams A., Nemec B., Ijspeert A. J., Ude A., 2014, ‘Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks’, IEEE Transactions on Robotics, 30, 4, pp. 816-830.

Ge S. S., Hang C. C., Woon L. C., 1997, ‘Adaptive Neural Network Control of Robot Manipulators in Task Space’, IEEE Transactions on Industrial Electronics, 44, 6.

Ghalyan I. F. J., 2016, ‘Force-Guided Robotic Assembly Process: Control and Contact-State Recognition’, PhD Thesis, Université du Luxembourg PhD-FSTC-206-1.

Ginoya D., Shendge P. D., Phadke S. B., 2015, ‘Disturbance observer based sliding mode control of nonlinear mismatched uncertain systems’, Communications in Nonlinear Science and Numerical Simulation, 26, pp. 98-107.

Giusti A., Zeestraten M. J. A., Icer E., Pereira A., Caldwell D. G., Calinon S., Althoff M., 2018, ‘Flexible Automation Driven by Demonstration: Leveraging Strategies that Simplify Robotics’, IEEE Robotics & Automation Magazine, 25, pp. 18-27.

Goodfellow I., Bengio Y., Courville A., 2016, ‘Deep Learning’, Cambridge: MIT Press.

Grosnan C., Abraham A., 2011, ´Rule-Based Expert Systems‘, Intelligent Systems Reference Library, 17, Berlin, Heidelberg: Springer-Verlag, pp. 149-185.

Gu S., Lillicrap T., Sutskever I., Levine S., 2016, ‘Continuous Deep Q-Learning with Model-based Acceleration’, arXiv: 1603.00748v1 [cs.LG].

Ha I.-C., Han M.-C., 2004, ‘Robust Hybrid Position/Force Control with Adaptive Scheme’, JSME International Journal Series C, 47, 4, pp. 1161-1165.

Hackl C. M., Endisch C., Schröder D., 2009, ‘Contributions to non-identifier based adaptive control in mechatronics‘, Robotics and Autonomous Systems, 57, 10, pp. 996-1005.

Hamdad L., Achab A., Boutouchent A., Dahamni F., 2013, ‘Self Organized Biogeography Algorithm for Clustering’, IWINAC 2013, Part I, LNCS 7930, pp. 396-405.

Hazara M., Kyrki V., 2017, ‘Reinforcement learning for improving imitated in-contact skills’, Proceedings – IEEE-RAS International Conference on Humanoid Robots, Humanoids 2016, pp. 194-201.

He W., Dong Y., 2018, ‘Adaptive Fuzzy Neural Network Control for a Constrained Robot Using Impedance Learning’, IEEE Transactions on Neural Networks and Learning Systems, 29, 4.

Hester T., Quinlan M., Stone P., 2011, ‘A Real-Time Model-Based Reinforcement Learning Architecture for Robot Control’, arXiv: 1105.1749v2 [cs.AI].

Hester T., Vecerik M., Pietquin O., Lanctot M., Schaul T., Piot B., Sendonaris A., Dulac-Arnold G., Osband I., Agapiou J., Leibo J. Z., Gruslys A., 2017, ‘Learning from Demonstrations for Real World Reinforcement Learning’, arXiv: 1704.03732v1 [cs.AI].

Huang L., Ge S. S., Lee T. H., 2003, ‘Fuzzy unidirectional force control of constrained robotic manipulators’, Fuzzy Sets and Systems, 134, pp. 135-146.

Huang S., Aertbeliën E., Bruyninckx H., Van Brussel H., 2013, ‘Behavior-based Task Learning by Demonstration for Mobile Manipulators’, International Journal of Automation and Smart Technology, 3, 1, pp. 19-28.

Huang Z., 2013, ‘A classification rules extraction algorithm base on fish swarm optimization’, International Journal of Engineering Science Invention, 2-4, pp. 31-33.

IEEE Robotics & Automation Society, Technical Committee for Robot Ethics, http://www.ieee-ras.org/robot-ethics, visited 20.07.2018.

Inasaki I., Karpuschewski B., Lee H.-S., 2001, ‘Grinding Chatter-Origin and Suppression’, CIRP Annals-Manufacturing Technology, 50, 2, pp. 515-534.

Jankowski K. P., ElMaraghy H. A., 1999, ‘Robust hybrid position/force control of redundant robots’, Robotics and Autonomous Systems, 27, pp. 111-127.

Jasim I. F., Plapper P. W., 2013, ‘Adaptive sliding mode control of switched constrained robotic manipulators’, Proceedings - 11th IEEE International Conference on Industrial Informatics (INDIN), pp. 305-310.

Jasim I. F., Plapper P. W., 2014, ‘Robust Direct Adaptive Fuzzy Control of Switched Constrained Manipulators with Unknown Dynamics’, ISR/ROBOTIK, pp. 170-177.

Jasim I. F., Plapper P. W., Voos H., 2015, ´Adaptive sliding mode fuzzy control for unknown robots with arbitrarily-switched constraints’, Mechatronics, 30, pp. 174-186.

Jiang J. L., Ge P. Q., Bi W. B., Zhang L., Wang D. X., Zhang Y., 2013, '2D / 3D ground surface topography modeling considering dressing and wear effects in grinding process', International Journal of Machine Tools & Manufacture, 74, pp. 29-40.

Johnson M., Vera A. H., 2019, 'No AI is an Island: The Case for Teaming Intelligence', AI Magazine Spring 2019, 40, 1, pp. 17-28.

Jonschkowski R., Hafner R., Scholz J., Riedmiller M., 2017, 'PVEs: Position-Velocity Encoders for Unsupervised Learning of Structured State Representations', arXiv: 1705.09805v3 [cs.RO].

Jung S., Hsia T. C., 2000, 'Robust neural force control scheme under uncertainties in robot dynamics and unknown environment', IEEE Transactions on Industrial Electronics, 47, 2, pp. 403-412.

Kaelbling L. P., Littman M. L., Moore A. W., 1996, 'Reinforcement Learning: A Survey', Journal of Artificial Intelligence Research, 4, pp. 237-285.

Kaihara T., Nishino N., Ueda K., Tseng M., Váncza J., Schönsleben P., Teti R., Takenaka T., 2018, 'Value creation in production: Reconsideration from interdisciplinary approaches', CIRP Annals-Manufacturing Technology, 67, pp. 791-813.

Kalakrishnan M., Righetti L., Pastor P., Schaal S., 2011, 'Learning Force Control Policies for Compliant Manipulation', Proceedings - IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4639-4644.

Kao Y., Li Y. L., 2008, 'Ant colony recognition systems for part clustering problems', International Journal of Production Research, 46, 15, pp. 4237-4258.

Kapotoglu M., Koc C., Sariel S., 2015, 'Robots Avoid Potential Failures through Experience-based Probabilistic Planning', Proceedings - 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), pp. 111-120.

Karayiannidis Y., Rovithakis G., Doulgeri Z., 2007, 'Force/position tracking for a robotic manipulator in compliant contact with a surface using neuro-adaptive control', Automatica, 43, pp. 1281-1288.

Karayiannidis Y., Doulgeri Z., 2010, 'Robot contact tasks in the presence of control target distortions', Robotics and Autonomous Systems, 58, 5, pp. 596-606.

Karayiannidis Y., Doulgeri Z., 2012, 'Model-free robot joint position regulation and tracking with prescribed performance guarantees', Robotics and Autonomous Systems, 60, 2, pp. 214-226.

Kawamura S., Miyazaki F., Arimoto S., 1988, 'Is a local linear PD feedback control law effective for trajectory tracking of robot motion?', Proceedings - IEEE International Conference on Robotics and Automation (ICRA), 3, pp. 1335-1340.

Khan S. G., Herrmann G., Lewis F. L., Pipe T., Melhuish C., 2012, 'Reinforcement learning and optimal adaptive control: An overview and implementation examples', Annual Reviews in Control, 36, pp. 42-59.

Khansari M., Klingbeil E., Khatib O., 2016, 'Adaptive human-inspired compliant contact primitives to perform surface-surface contact under uncertainty', The International Journal of Robotics Research, 35, 13, pp. 1651-1675.

Kiguchi K., Fukuda T., 2000, 'Position/Force Control of Robot Manipulators for Geometrically Unknown Objects Using Fuzzy Neural Networks', IEEE Transactions on Industrial Electronics, 47, 3, pp. 641-649.

Kiguchi K., Watanabe K., Fukuda T., 2002, 'Generation of efficient adjustment strategies for a fuzzy-neuro force controller using genetic algorithms – application to robot force control in an unknown environment', Information Sciences, 145, pp. 113-126.

Klecker S., Plapper P. W., 2016, ´Adaptive SMC For Trajectory Tracking In FreeForm Grinding‘, Proceedings – IEEE International Conference on Industrial Informatics (INDIN), pp. 196-201.

Klecker S., Plapper P., 2016b, 'Trajectory Tracking for Robotic Freeform Grinding', Cahier Scientifique – Revue Technique Luxembourgeoise, 1, pp. 6-7.

Klecker S., Plapper P., 2016c, 'BELBIC-Sliding Mode Control of Robotic Manipulators with Uncertainties and Switching Constraints', Proceedings – ASME International Mechanical Engineering Congress and Exposition (IMECE).

Klecker S., Hichri B., Plapper P., 2017, ´Robust BELBIC-Extension for Trajectory Tracking Control‘, Journal of Mechanics Engineering and Automation, 7, 2, pp. 84-93.

Klecker S., Plapper P., Hichri B., 2017b, 'PID and Biomimetic Variable Structure Path Tracking Control in Automated Surface Finishing Processes', Proceedings – Robotix-Academy Conference for Industrial Robotics (RACIR), pp. 18-24.

Klecker S., Hichri B., Plapper P., 2017c, ´Neuro-Inspired Reward-Based Tracking Control for Robotic Manipulators with Unknown Dynamics', Proceedings – International Conference on Robotics and Automation Engineering, pp. 21-25.

Klecker S., Hichri B., Plapper P., 2018, ´CILAP-Architecture for Simultaneous Position- and Force-Control in Constrained Manufacturing Tasks', Proceedings – International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2, pp. 244-251.

Klecker S., Hichri B., Plapper P., 2018b, ´Learning-While Controlling RBF-NN for Robot Dynamics Approximation in Neuro-Inspired Control of Switched Nonlinear Systems', Lecture Notes in Computer Science, 11141, Artificial Neural Networks and Machine Learning, ICANN, 3, pp. 717-727.

Klecker S., Hichri B. Plapper P., 2019, ´Rule-Based Supervisory Control-Extension for Automated Manufacturing Processes', International Journal of Mechanical Engineering and Robotics Research.

Klocke F., Blattner M., Adams O., Brockmann M., Veselovac D., 2014, 'Compensation of Disturbances on Force Signals for Five-Axis Milling Processes', Procedia CIRP, 14, pp. 472-477.

Klocke F., König W., 2005, 'Fertigungsverfahren 2-Schleifen, Honen, Läppen', Berlin, Heidelberg: Springer-Verlag.

Kober J., Gienger M., Steil J. J., 2015, 'Learning Movement Primitives for Force Interaction Tasks‘, Proceedings - IEEE International Conference on Robotics and Automation (ICRA), pp. 3192-3199.

Kober J., Oztop E., Peters J., 2011, 'Reinforcement Learning to Adjust Robot Movements to New Situations', Robotics: Science and Systems (RSS).

Koenig N., Matarić M. J., 2017, 'Robot life-long task learning from human demonstrations: a Bayesian approach‘, Autonomous Robots, 41, 5, pp.1173-1188.

Kormushev P., Calinon S., Caldwell D. G., 2011, 'Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input', Advanced Robotics, 25, 5, pp. 581-603.

Krabbes M., Döschner C., 1999, 'Modelling of Robot Dynamics based on a Multi-dimensional RBF-like Neural Network', Proceedings - IEEE International Conference on Information, Intelligence, and Systems.

Kramberger A., Gams A., Nemec B., Chrysostomou D., Madsen O., Ude A., 2017, 'Generalization of orientation trajectories and force-torque profiles for robotic assembly', Robotics and Autonomous Systems, 98, pp. 333-346.

Kronander K., Billard A., 2014, ‘Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction’, IEEE Transactions on Haptics, 7, 3, pp. 367-380.

Kuc T., Han W., 2000, ‘An adaptive PID learning control of robot manipulators.’, Automatica, 36, pp. 717-725.

KUKA, 2011, ‘KUKA_LBR4plus_ENGLISCH’ via https://www.kukakore.com/wp-content/uploads/2012/07/KUKA_LBR4plus_ENLISCH.pdf, visited 25.03.2019.

KUKA, 2012, ‘KUKA System Software 5.6 lr-Operating and Programming Instructions for System Integrators-Version: KSS 5.6 lr SI V5 en’, Pub KSS 5.6 lr SI (PDF) en, Augsburg: KUKA Laboratories GmbH.

KUKA, 2017, ‘KUKA.CAMRob KRC V3.0 Option to KSS 8.3’, KST_CAMRob_KRC_30_EN.docx, Houthalen-Helchteren: KUKA Automatisering+Robots N.V..

Kumar N., Panwar V., Sukavanam N., Sharma S. P., Borm J.-H., 2011, ‘Neural Network Based Hybrid Force/Position Control for Robot Manipulators’, International Journal of Precision Engineering and Manufacturing, 12, 3, pp. 419-426.

Kurfess T. R., Whitney D. E., Brown M. L., 1988, ‘Verification of a Dynamic Grinding Model’, Journal of Dynamic Systems, Measurement, and Control.

Lange F., Bertleff W., Suppa M., 2013, ‘Force and Trajectory Control of Industrial Robots in Stiff Contact’, Proceedings – IEEE International Conference on Robotics and Automation (ICRA), pp. 2927-2934.

LeCun Y., Bengio Y., Hinton G., 2015, ‘Deep learning’, Nature, 521, pp. 436-444.

Lee A. X., Lu H., Gupta A., Levine S., Abbeel P., 2015, ‘Learning Force-Based Manipulation of Deformable Objects from Multiple Demonstrations’, Proceedings – IEEE International Conference on Robotics and Automation (ICRA), pp. 177-184.

Lee M. C., Go S. J., Lee M. H., Jun C. S., Kim D. S., Cha K. D., Ahn J. H., 2001, ‘Robust trajectory tracking control of a polishing robot system based on CAM data', Robotics and Computer-Integrated Manufacturing, 17, 1, 2, pp. 177-183.

Lee M.-J., Choi Y.-K., 2004, ‘An Adaptive Neurocontroller Using RBFN for Robot Manipulators’, IEEE Transactions on Industrial Electronics, 51, 3, pp. 711-717.

Lewis F. L., Liu K., Yesildirek A., 1995, ‘Neural Net Robot Controller with Guaranteed Tracking Performance’, IEEE Transactions on Neural Networks, 6, 3, pp. 703-715.

Lewis F. L., Yesildirek A., Liu K., 1996, ‘Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance’, IEEE Transactions on Neural Networks, 7, 2, pp. 388-399.

Lewis F. L., Vrabie D., 2009, ‘Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control’, IEEE Circuits and Systems Magazine, 9, 3, pp. 32-50.

Li T. S., Huang Y., 2010, ‘MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators’, Information Sciences, 180, 23, pp. 4641-4660.

Li Y., 2018, ‘Deep Reinforcement Learning: An Overview’, arXiv: 1701.07274v6 [cs.LG].

Li Y., Ge S. S., Zhang Q., Lee T. H., 2013, ‘Neural networks impedance control of robots interacting with environments‘, IET Control Theory and Applications, 7, 11, pp. 1509-1519.

Li Y., Yang G., 2015, ‘Robust fuzzy adaptive fault-tolerant control for a class of nonlinear systems with mismatched uncertainties and actuator faults’, Nonlinear Dynamics 81, pp. 395-409.

Lian R.-J., 2013, ‘Enhanced adaptive grey-prediction self-organizing fuzzy sliding-mode controller for robotic systems’, Information Sciences, 236, pp. 186-204.

Liberzon D., 2003, ‘Switching in Systems and Control’, Boston: Birkhäuser.

Lin P., Abney K., Bekey G., 2011, ‘Robot ethics: Mapping the issues for a mechanized world’, Artificial Intelligence, 175, pp. 942-949.

Lin S.-T., Lee J.-S., 1996, ‘Impedance Control with On-Line Neural-Network Compensator for Robot Contact Tasks‘, Journal of Intelligent and Robotic Systems, 15, pp. 389-399.

Lin W., Xu P., Li B., Yang, X., 2014, ‘Path planning of mechanical polishing process for freeform surface with a small polishing tool’, Robotics and Biomimetics, 1, 1.

Liu H., Fang T., Zhou T., Wang Y., Wang L., 2018, ‘Deep learning-based Multimodal Control Interface for Human-Robot Collaboration‘, Procedia CIRP, 72, pp. 3-8.

Liu J., Wang X., 2011, ‘Advanced Sliding Mode Control for Mechanical Systems: Design, Analysis and MATLAB Simulation’, Beijing, Heidelberg: Tsinghua University Press and Springer-Verlag.

Liu J., 2013, ‘Radial Basis Function (RBF) Neural Network Control for Mechanical Systems’, Beijing, Heidelberg: Tsinghua University Press and Springer-Verlag.

Lucas C., Shahmirzadi D., Sheikholeslami N., 2004, ‘Introducing Belbic : Brain Emotional Learning Based Intelligent Controller’, Intelligent Automation and Soft Computing, 10, 1, pp. 11-22.

Luh J. Y. S., Walker M. W., Paul R. P. C., 1980, ‘Resolved-Acceleration Control of Mechanical Manipulators’, IEEE Transactions on Automatic Control, AC-25, 3, pp. 468-474.

Markiewicz B. R., 1973, ‘Analysis of the Computed Torque Drive Method and Comparison With Conventional Position Servo for a Computer-Controlled Manipulator’, Technical Memorandum 33-601 Jet Propulsion Laboratory California Institute of Technology Pasadena California.

Martínez D., Alenyà G., Torras C., 2017, ‘Relational reinforcement learning with guided demonstrations’, Artificial Intelligence, 247, pp. 295-312.

Matarić M. J., 1998, ‘Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior’, Trends in Cognitive Sciences, 2, 3, pp. 82-87.

Mayr O., 1970, ‘The Origins of Feedback Control’, Cambridge, London: The M.I.T. Press.

McCulloch W. S., Pitts W., 1943, ‘A logical calculus of the ideas immanent in nervous activity’, Bulletin of Mathematical Biophysics, 5, pp. 115-133.

Merrick, K. E., 2012, ‘Intrinsic Motivation and Introspection in Reinforcement Learning.’, IEEE Transactions on Autonomous Mental Development, 4, 4, pp. 315–329.

Micchelli, C. A., 1986, ‘Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions’, Constructive Approximation, 2, pp. 11-22.

Mishra A., Zaheeruddin, 2010, ‘Design of Fuzzy Neural Network for Function Approximation and Classification’, IAENG International Journal of Computer Science, 37, 4, pp. 326-340.

Moghadam H. Z., Landers R. G., Balakrishnan S. N., 2014, ‘Hierarchical optimal contour control of motion systems’, Mechatronics, 24, pp. 98-107.

Moghadam H. Z., Landers R. G., Balakrishnan S. N., 2014b, ‘Hierarchical optimal force-position control of complex manufacturing processes’, Control Engineering Practice, 25, pp. 75-84.

Mohd Adnan M. R. H., Sarkheyli A., Mohd Zain A., Haron H., 2015, ‘Fuzzy logic for modeling machining process: a review’, Artificial Intelligence Review, 43, 3, pp. 345-379.

Montebelli A., Steinmetz F., Kyrki V., 2015, ‘On Handing Down Our Tools to Robots: Single-Phase Kinesthetic Teaching for Dynamic In-Contact Tasks‘, Proceedings – IEEE International Conference on Robotics and Automation (ICRA), pp. 5628-5634.

Morén J., Balkenius C., 2000, ‘A computational model of emotional learning in the amygdala’, Proceedings - 6th International Conference on Simulation of Adaptive Behavior, 32.

Navarro-Gonzalez J. L., Lopez-Juarez I., Rios-Cabrera R., Ordaz-Hernández K., 2015, ‘On-line knowledge acquisition and enhancement in robotic assembly tasks’, Robotics and Computer-Integrated Manufacturing, 33, pp. 78-89.

Newman W. S., Branicky M. S., Podgurski H. A., Chhatpar S., Huang L., Swaminathan J., Zhang H., 1999, ‘Force-responsive robotic assembly of transmission components’, Proceedings - IEEE International Conference on Robotics and Automation (ICRA), 3, pp. 2096-2102.

Ng A. Y., Russell S., 2000, ‘Algorithms for Inverse Reinforcement Learning‘, Proceedings - 7th International Conference on Machine Learning, pp. 663-670.

Noroozi N., Roopaei M., Jahromi M. Z., 2009, ´Adaptive fuzzy sliding mode control scheme for uncertain systems’, Communications in Nonlinear Science and Numerical Simulation, 14, 11, pp. 3978-3992.

Oba Y., Yamada Y., Igarashi K., Katsura S., Kakinuma Y., 2016, ‘Replication of skilled polishing technique with serial-parallel mechanism polishing machine’, Precision Engineering, 45, pp. 292-300.

Octopuz, http://octopuz.com/, visited 28.06.2016.

Ortega R., Loria A., Kelly R., 1995, ‘A Semiglobally Stable Output Feedback PI$^2$D Regulator for Robot Manipulators’, IEEE Transactions on Automatic Control, 40, 8, pp. 1432-1436.

Pagilla P. R., Yu B., 2001, ‘Adaptive control of robotic surface finishing processes’, Proceedings-American Control Conference, pp. 630-635.

Park J., Kim S. H., Kim S., 2008, ‘Active Compliant Motion Control for Grinding Robot‘, Proceedings - 17th World Congress The International Federation of Automatic Control (IFAC), 17, pp. 4285-4289.

Park J., Sandberg I. W., 1991, ‘Universal Approximation Using Radial-Basis-Function Networks‘, Neural Computation, 3, pp. 246-257.

Passino K. M., 2005, ‘Biomimicry for Optimization, Control, and Automation’, London: Springer-Verlag.

Peng W., Lin Z., Su J., 2008, ‘Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques’, IET Control Theory and Applications, 3, 6, pp. 701-711.

Perzylo A., Rickert M., Kahl B., Somani N., Lehmann C., Kuss A., Profanter S., Beck A. B., Haage M., Hansen M. R., Nibe M. T., Roa M. A., Sörnmo O., Robertz S. G., Thomas U., Veiga G., Topp E. A., Kessler I., Danzer M., 2019, ‘SMErobotics – Smart Robots for Flexible Manufacturing’, IEEE Robotics & Automation Magazine, DOI: 10.1109/MRA.2018.2879747, pp. 78-90.

Peters J., Schaal S., 2008, ‘Reinforcement learning of motor skills with policy gradients’, Neural Networks, 21, pp. 682-697.

Phillips M., Hwang V., Chitta S., Likhachev M., 2016, ‘Learning to plan for constrained manipulation from demonstrations’, Autonomous Robots, 40, 1, pp. 109-124.

Pliego-Jiménez J., Arteaga-Pérez M. A., 2015, ‘Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters’, European Journal of Control, 22, pp. 1-12.

Polydoros A. S., Nalpantidis L., 2017, ‘Survey of Model-Based Reinforcement Learning: Applications on Robotics’, Journal of Intelligent and Robotic Systems: Theory and Applications, 86, 2, pp. 153-173.

Poole D. L., Mackworth A. K., 2017, ‘Artificial Intelligence: Foundations of Computational Agents, 2nd Edition’, Cambridge: Cambridge University Press.

Qu Z., Dorsey J. F., Zhang X., Dawson D. M., 1991, ‘Robust control of robots by the computed torque law’, Systems & Control Letters, 16, 1, pp. 25-32.

Qu Z., Dorsey J., 1991, ‘Robust PID control of Robots’, International Journal of Robotics and Automation, 6, pp. 228-235.

Qu Z., Dawson D. M., Dorsey J. F., Duffie J. D., 1995, ‘Robust estimation and control of robotic manipulators’, Robotica, 13, 3, pp. 223-231.

Qin J., Léonard F., Abba G., 2016, ‘Real-Time Trajectory Compensation in Robotic Friction Stir Welding Using State Estimators’, IEEE Transactions on Control Systems Technology, 24, 6, pp. 2207-2214.

Racca M., Pajarinen J., Montebelli A., Kyrki V., 2016, ‘Learning In-Contact Control Strategies from Demonstration’, Proceedings - IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 688-695.

Ren X., Rad A. B., Lewis F. L., 2007, ‘Neural Network-Based Compensation Control of Robot Manipulators with Unknown Dynamics‘, Proceedings - American Control Conference, pp. 13-18.

Rey J., Kronander K., Farshidian F., Buchli J., Billard A., 2018, ‘Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies’, Autonomous Robots, 42, 1, pp. 45-64.

Reynolds C. W., 1987, ‘Flocks, Herds, and Schools: A Distributed Behavioral Model’, Computer Graphics, 21, 4.

Riedmiller M., Hafner R., Lampe T., Neunert M., Degrave J., Van de Wiele T., Mnih V., Heess N., Springenberg T., 2018, ‘Learning by Playing-Solving Sparse Reward Tasks from Scratch‘, arXiv:1802.10567v1 [cs.LG].

Robertsson A., Olsson T., Johansson R., Blomdell A., Nilsson K., Haage M., Lauwers B., de Baerdemaeker H., Brogardh T., Brantmark H., 2006, ‘Implementation of Industrial Robot Force Control Case Study: High Power Stub Grinding and Deburring’, Proceedings - IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2743–2748.

Robotmaster, http://www.robotmaster.com/en/, visited 28.06.2016.

Roopaei M., Zolghadri Jahromi M., 2009, ‘Chattering-free fuzzy sliding mode control in MIMO uncertain systems’, Nonlinear Analysis, 71, 10, pp. 4430-4437.

Roswell A., Jeff F., Liu G., 2006, ‘Modelling and analysis of contact stress for automated polishing’, International Journal of Machine Tools & Manufacture, 46, pp. 424–435.

Rozo L., Jiménez P., Torras C., 2013, ‘A robot learning from demonstration framework to perform force-based manipulation tasks’, Intelligent Service Robotics, 6, pp. 33–51.

Rozo L., Silvério J., Calinon S., Caldwell D. G., 2016, ‘Learning Controllers for Reactive and Proactive Behaviors in Human-Robot Collaboration’, Frontiers in Robotics and AI, 3, pp. 1-11.

Scheutz M., 2013, ‘What is Robot Ethics?’, IEEE Robotics & Automation Magazine.

Schreiber G., 2010, ‘KUKA New Connectivity: Fast Research Interface (FRI) for KRC2LR „Get FRI – become free (to research)"’, Augsburg: KUKA Roboter GmbH.

Shah H., Chakravarty S., 2009, ‘Kinematic and Dynamic Control of a Two Link Manipulator’, via https://nl.mathworks.com/matlabcentral/fileexchange/23963-kinematic-dynamic-control-of-a-two-link-manipulator, visited 11.03.2019.

Sharbafi M. A., Lucas C., Daneshvar R., 2010, ‘Motion Control of Omni-Directional Three-Wheel Robots by Brain-Emotional-Learning-Based Intelligent Controller’, IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 40, 6, pp. 630-638.

Siciliano, B., Khatib, O., 2008, ‘Springer Handbook of Robotics‘, Berlin, Heidelberg: Springer-Verlag.

Siciliano B., Sciavicco L., Villani L., Oriolo G., 2010, ‘Robotics Modelling, Planning and Control’, London: Springer-Verlag.

Siciliano B., Villani L., 1996, ‘A Passivity-based Approach to Force Regulation and Motion Control of Robot Manipulators’, Automatica, 32, 3, pp. 443-447.

Silver D., Schrittwieser J., Simonyan K., Antonoglou I., Huang A., Guez A., Hubert T., Baker L., Lai M., Bolton A., Chen Y., Lillicrap T., Hui F., Sifre L., van den Driessche G., Graepel T., Hassabis D., 2017, ‘Mastering the game of Go without human knowledge’, Nature, 550, pp. 354-359.

Singh H. P., Sukavanam N., 2013, ‘Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties’, Neural Computing and Applications, 22, pp. 1745-1755.

Skoglund A., Iliev B., Palm R., 2010, ‘Programming-by-Demonstration of reaching motions – A next-state-planner approach’, Robotics and Autonomous Systems, 58, pp. 607-621.

Skubic M., Volz R.A., 2000, ‘Acquiring Robust, Force-Based Assembly Skills from Human Demonstration’, IEEE Transactions on robotics and automation, 16, 6, pp. 772-781.

Slotine J. E., Li W., 1991, ‘Applied Nonlinear Control’, Upper Saddle River: Prentice Hall.

Spong M. W., Hutchinson S., Vidyasagar M., 2006, ‘Robot Modeling and Control’, New York: Wiley.

Stepien P., 2009, ‘A probabilistic model of the grinding process’, Applied Mathematical Modelling, 33, pp. 3863-3884.

Strehl A. L., Li L., Wiewiora E., Langford J., Littman M. L., 2006, ‘PAC Model-Free Reinforcement Learning‘, Proceedings - 23rd International Conference on Machine Learning (ICML), pp. 881-888.

Sun F. C., Sun Z. Q., 1997, ‘Stable sampled-data adaptive control of robot arms using neural networks’, Journal of Intelligent and Robotic Systems, 20, 4, pp. 131-155.

Sun F. C., Sun Z. Q., Feng G., 1999, ‘An Adaptive Fuzzy Controller Based on Sliding Mode for Robot Manipulators’, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 29, 4, pp. 661-667.

Sun F. C., Sun Z. Q., Zhang R. J., Chen Y. B., 2000, ‘Neural adaptive tracking controller for robot manipulators with unknown dynamics’, IEEE Proceedings-Control Theory and Applications, 147, 3, pp. 366-370.

Sun T., Pei H., Pan Y., Zhou H., Zhang C., 2011, ‘Neural network-based sliding mode adaptive control for robot manipulators’, Neurocomputing, 74, pp. 2377-2384.

Sünderhauf N., Brock O., Scheirer W., Hadsell R., Fox D., Leitner J., Upcroft B., Abbeel P., Burgard W., Milford M., Corke P., 2018, ‘The limits and potentials of deep learning for robotics‘, The International Journal of Robotics Research, 37, 4, 5, pp. 405-420.

Sutton R. S., Barto A. G., 1998, ‘Reinforcement Learning – An Introduction’, Cambridge, London: A Bradford Book, The MIT Press.

Szabo A., De Castro L. N., Delgado M. R., 2012, ‘A Constructive Particle Swarm Algorithm for Fuzzy Clustering’, IDEAL 2012, LNCS 7435, pp. 390-398.

Tabot G.A., Kim S.S., Winberry J.E., Bensmaia S.J., 2014, ‘Restoring tactile and proprioceptive sensation through a brain interface’, Neurobiology of Disease.

Tao Y., Zheng J., Lin Y., 2016, ‘A Sliding Mode Control-based on a RBF Neural Network for Deburring Industry Robotic Systems’, International Journal of Advanced Robotic Systems, 13, 1.

Toibero J. M., Roberti F., Carelli R., Fiorini P., 2011, ‘Switching control approach for stable navigation of mobile robots in unknown environments’, Robotics and Computer-Integrated Manufacturing, 27, pp. 558-568.

Tonkens R., 2009, ‘A Challenge for Machine Ethics’, Minds & Machines, 19, pp. 421-438.

Touchlab UCL, 2014, via https://vr.cs.ucl.ac.uk/wp-content/uploads/2014/05/KUKA_LWR4.png, visited 09.04.2019.

Tramper J.J., Flanders M., 2013, ‘Predictive mechanisms in the control of contour following’, Experimental Brain Research, 227, 4, pp. 535-546.

Unbehauen H., 2008, ‘Regelungstechnik 1’, Wiesbaden: Vieweg+Teubner Verlag|GWV Fachverlage GmbH.

Vaijayanthi R. P., Natarajan A. M., Murugadoss J. R., 2011, ‘Document Clustering using Hybrid ACO-TS’, International Journal of Recent Trends in Engineering and Technology, 6, 1.

Van Cuong P., Nan W. Y., 2016, ‘Adaptive trajectory tracking neural network control with robust compensator for robot manipulators’, Neural Computing and Applications, 27, 2, pp. 525-536.

Veruggio G., Operto F., 2008, ’64 Roboethics: Social and Ethical Implications of Robotics’ in B. Siciliano, O. Khatib, ‘Springer Handbook of Robotics’, Berlin, Heidelberg: Springer-Verlag, pp. 1499-1524.

Villani V., Pini F., Leali F., Secchi C., 2018, ‘Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications’, Mechatronics, 55, pp. 248-266.

Visser L. C., Carloni R., ünal R., Stramigioli S., 2010, ‘Modeling and Design of Energy Efficient Variable Stiffness Actuators’, Proceedings - IEEE International Conference on Robotics and Automation (ICRA), 1, pp. 3273-3278.

Von Wattenwyl M., Clerici M., Brauchli H., 2001, ‘Independent Hybrid Force/Motion Control of Constrained Six-Degrees-of-Freedom Manipulators‘, Multibody System Dynamics, 6, pp. 327-342.

Wallach W., Allen C., 2010, ‘Moral Machines-Teaching Robots Right from Wrong’, Oxford: Oxford University Press.

Wang H., 2011, ‘On adaptive inverse dynamics for free-floating space manipulators’, Robotics and Autonomous Systems, 59, pp. 782-788.

Wang L., Chai T., Zhai L., 2009, ‘Neural-Network-Based Terminal Sliding-Mode Control of Robotic Manipulators Including Actuator Dynamics’, IEEE Transactions on Industrial Electronics, 56, 9, pp. 3296-3304.

Weinert K., Blum H., Jansen T., Rademacher A., 2007, ´Simulation based optimization of the NC-shape grinding process with toroid grinding wheels‘, Production Engineering, 1, pp. 245-252.

Wen J. T., Murphy S. H., 1990, ‘PID Control for Robot Manipulators’, Center for Intelligent Robotic Systems for Space Exploration, Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, New York, CIRSSE Document #54.

Wilbert A. D., Behrens B., Dambon O., Klocke F., 2012, ‘Robot Assisted Manufacturing System for High Gloss Finishing of Steel Molds State of the Art’, ICIRA 2012, Part I, LNAI 7506, pp. 673–685.

Wyatt I. D., Hecker D. E., 2006, ‘Occupational changes during the 20th century‘, Monthly Labor Review, 35.

Xu W., Minami M., Mae Y., 2007, ‘Position/Force Control of Grinding Robot by Using Real-time Presumption of Constrained Condition’, Proceedings - SICE Annual Conference, pp. 1861-1868.

Xu W., Cai Chenxiao, Zou Y., 2014, ‘Neural-network-based robot time-varying force control with uncertain manipulator environment’, Transactions of the Institute of Measurement and Control, 36, 8, pp. 999-1009.

Yamaguchi A., Atkeson C. G., 2016, ‘Neural Networks and Differential Dynamic Programming for Reinforcement Learning Problems’, Proceedings – IEEE International Conference on Robotics and Automation (ICRA), 1, pp. 5434-5441.

Yang C., Zeng C., Cong Y., Wang N., Wang M., 2018, ‘A Learning Framework of Adaptive Manipulative Skills from Human to Robot’, IEEE Transactions on Industrial Informatics, DOI 10.1109/TII.2018.2826064.

Yao B., Tomizuka M., 1998, ‘Adaptive Robust Motion and Force Tracking Control of Robot Manipulators in Contact With Compliant Surfaces With Unknown Stiffness’, Journal of Dynamic Systems, Measurement, and Control, 120, 2, DOI:10.1115/1.2802414.

Yi H., 2015, ‘A Sliding Mode Control Using Brain Limbic System Control Strategy for a Robotic Manipulator’, International Journal of Advanced Robotic Systems, 12, 158.

Yin Y., Hu H., Xia Y., 2004, ‘Active tracking of unknown surface using force sensing and control technique for robot’, Sensors and Actuators, A 112, pp. 313-319.

Yin Y. H., Xu Y., Jiang Z. H., Wang Q. R., 2012, ‘Tracking and Understanding Unknown Surface With High Speed by Force Sensing and Control for Robot’, IEEE Sensors Journal, 12, 9, pp. 2910-2916.

Yixu S., Hongbo L., Zehong Y., 2012, ‘An adaptive modeling method for a robot belt grinding process’, IEEE/ASME Transactions on Mechatronics, 17, 2, pp. 309–317.

You S.-S., 1996, ‘A Unified Dynamic Model and Control Synthesis for Robotic Manipulators with Geometric End-Effector Constraints’, KSME Journal, 10, 2, pp. 203-212.

Yu L., Fei S., Huang J., Yu G., 2014, ‘Trajectory Switching Control of Robotic Manipulators Based on RBF Neural Networks’, Circuits, Systems and Signal Processing, 33, 4, pp. 1119-1133.

Zhang J., Ferch M., 2003, ‘Extraction and transfer of fuzzy control rules for sensor-based robotic operations’, Fuzzy Sets and Systems, 134, pp. 147-167.

Zhang X., Polydoros A. S., Piater J., 2018, ‘Learning Movement Assessment Primitives for Force Interaction Skills’, arXiv: 1805.04354v1 [cs.RO].

Ziauddin S. M., Zalzala A. M. S., 1994, ‘Neuro-adaptive hybrid position/force control of robotic manipulators‘, Robotics Research Group, Department of Automatic Control and Systems Engineering, University of Sheffield, Research Report no. 543.

## List of Publications

Klecker S., Hichri B., Plapper P., 2019, ´Rule-Based Supervisory Control-Extension for Automated Manufacturing Processes‘, International Journal of Mechanical Engineering and Robotics Research - European Conference on Materials, Mechatronics and Manufacturing (ECMMM 2019), Amsterdam, The Netherlands, 16.02.2019 - 18.02.2019 (Best presentation award).

Klecker S., Hichri B., Plapper P., 2018, ´Learning-While Controlling RBF-NN for Robot Dynamics Approximation in Neuro-Inspired Control of Switched Nonlinear Systems’, Lecture Notes in Computer Science 11141, Artificial Neural Networks and Machine Learning ICANN 2018 part 3 – 27th International Conference on Artificial Neural Networks (ICANN 2018), Rhodes, Greece, 04.10.2018 – 07.10.2018, pp. 717-727.

Klecker S., Hichri B., Plapper P., 2018, ´CILAP-Architecture for Simultaneous Position- and Force-Control in Constrained Manufacturing Tasks‘, Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics 2 (ICINCO 2018), Porto, Portugal, 29.07.2018 – 31.07.2018, pp. 244-251.

Klecker S., Hichri B., Plapper P., 2017, ‘Neuro-Inspired Reward-Based Tracking Control for Robotic Manipulators with Unknown Dynamics‘, Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, China, 29.12.2017 – 31.12.2017, pp. 21-25.

Klecker S., Plapper P., Hichri B., 2017, ´PID and Biomimetic Variable Structure Path Tracking Control in Automated Surface Finishing Processes‘,Robotix-Academy Conference for Industrial Robotics (RACIR) 2017, Rainer Müller, Peter Plapper, Olivier Brüls, Wolfgang Gerke, Gabriel Abba, Bassem Hichri and Matthias Vette-Steinkamp (Hrsg.), Berichte aus der Robotik, Aachen: Shaker Verlag, pp. 18-24 - RACIR, Luxembourg, Luxembourg, 06.06.2017 – 07.06.2017.

Klecker S., Hichri B., Plapper P., 2017, ´Robust BELBIC-Extension for Trajectory Tracking Control‘, Journal of Mechanics Engineering and Automation, 7, 2, pp. 84-93.

Klecker S., Plapper P., 2016, ‘Trajectory Tracking for Robotic Freeform Grinding’, Cahier Scientifique – Revue Technique Luxembourgeoise, pp. 6-7.

Klecker S., Plapper P., 2016, ‘BELBIC-Sliding Mode Control of Robotic Manipulators with Uncertainties and Switching Constraints’, Proceedings of the ASME 2016 International Mechanical Engineering Congress and Exposition (IMECE), Phoenix, USA, 11.11.2016 – 17.11.2016.

Klecker S., Plapper P. W., 2016, ‘Adaptive SMC For Trajectory Tracking In FreeForm Grinding’, Proceedings of the 14th IEEE International Conference on Industrial Informatics (IEEE-INDIN 2016), Poitiers, France, 18.07.2016 – 21.07.2016, pp. 196-201.

Robotix-Academy Conference for Industrial Robotics (RACIR) 2019: ´ Robotic assistants in factory routines – the ethical implications´ (accepted).

13th CIRP Conference on Intelligent Computation in Manufacturing Engineering: ‘Robotic trajectory tracking: Bio-inspired position and torque control’ (accepted).

## List of Supervised Student Projects

Diedrich, Gilbert:
´Mensch-Roboter-Interaktion‘,
Bachelor Thesis,
Bachelor Professionnel en Ingénierie – Mécanique Générale,
08/2018 – 02/2019.

Feller, Martin:
´Implementation of a PID Controller : Case Study‘,
Master Advanced Project / Case Study,
Master of Science in Engineering – Sustainable Product Creation,
08/2018 – 01/2019.

Daneshian, Sina:
´Simulation of a controller on the "KUKA LWR4+" in ROS’,
Master Advanced Project / Case Study,
Master of Science in Engineering – Sustainable Product Creation,
11/2017 – 02/2018.

Paola Hernández:
‘Gripper-design and CAD/CAM-robot programming’,
Student worker,
Master of Science in Engineering – Sustainable Product Creation,
04/2016 - 04/2017.