Semi-parametric identification of manipulator dynamics in a time-varying environment

N.L.D. Marck

Master of Science Thesis

Delft Center for Systems and Control

Semi-parametric identification of manipulator dynamics in a time-varying environment

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

N.L.D. Marck

March 28, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

A recent trend in robotics is aimed at the cooperation between human and robot. This has led to an increased development of collaborative robot manipulators. Typical characteristics of collaborative robots are their user-friendly and lightweight design, innovative compliant mechanics, the implementation of various safety features and advanced control capabilities. These characteristics enable humans to work alongside the manipulator or interact with it.

The implementation of passive compliant components such as springs and pneumatics have a beneficial effect on the level of safety for the operator. However, the added complexity often has a negative influence on the degree to which an accurate description of the system dynamics can be derived. Furthermore, the lightweight design and increasing payload-to-weight ratio amplify the effect of exogenous alterations to the system, such as attaching an object to the end effector.

The work in this thesis is aimed at obtaining an accurate description of the system dynamics for control purposes. In doing so, special attention is given to dealing with instantaneous time-varying phenomena.

An online semi-parametric approach is used to produce a valid description of the inverse dynamics of the considered system. The method consists of a non-parametric part which is described using Gaussian process regression (GPR) and a parametric part for which the parameters are identified using an extended Kalman filter (EKF). In this thesis, instantaneous system changes are introduced by attaching an unknown object to the end effector of the manipulator. The EKF implementation is specifically aimed at rapidly compensating for the response induced by this object. The GPR is used to compensate for remaining modeling errors.

The performance of the proposed methods is evaluated in simulation. The semi-parametric description achieves high modeling accuracy, fast adaptation to instantaneous system changes and reasonable generalization capabilities. Implementing the proposed solution in real-time applications requires additional research on the subject of online GPR.

Table of Contents

Acknowledgements

I would like to thank my supervisor Dr.-Ing. Jens Kober for his assistance during the writing of this thesis. I especially appreciate that Jens was always prepared to reserve time for a meeting, even on short notice. During our meetings he has provided many useful insights and suggestions.

Additionally, I want to thank Bart Hazen and Piet Aarden for their support. We have followed a similar path during our time at the university and I have enjoyed our collaborations on a technical, but also on a personal level.

Furthermore, I would like thank my parents and other close relatives for encouraging me during my time at the TU Delft. Special thanks go out to my girlfriend Linda van der Kleij, who has been involved with my thesis from the beginning to the end.

Delft, University of Technology N.L.D. Marck
March 28, 2017

Chapter 1

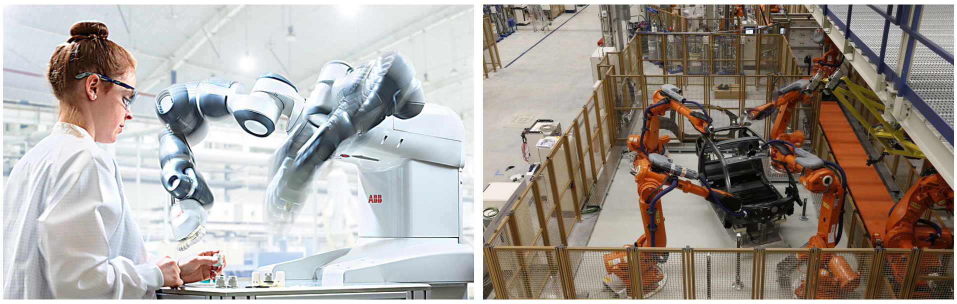
Introduction

Robotic manipulators are increasingly being integrated in various industries, such as the automotive, electronics, chemical and food industries [1, 2]. Initially, deploying robotic manipulators was only a possibility for large companies with high production rates. Now, the use of such manipulators is spreading towards smaller and more specialized companies as well [2, 3]. The demand for robotic manipulators is predicted to continue to rise in the coming years [1]. With this forecast in mind, an increase in research and development is essential.

An important topic that gets more attention is the development of collaborative robots [4]. Due to the use of innovative compliant designs for such robots, accurate knowledge of the system dynamics becomes more difficult to obtain. Nonetheless, the availability of this system knowledge is essential for cooperative control methods. This thesis is concerned with resolving this dilemma. In Section 1-1, background information is given on collaborative industrial robot manipulators. Section 1-2 continues with the objectives of this thesis. Finally, in Section 1-3, the outline of the remainder of this thesis is given.

1-1 Collaborative robot manipulators

To clarify the concept of collaborative industrial robots, note that ISO 8373 defines an industrial robot as “An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications.” This rather wide definition indicates the versatility of such devices. A *collaborative* industrial robot qualifies as a subclass of industrial robots, and refers to manipulators that are specifically designed to facilitate cooperation between humans and robots [5]. The definition of such a robot is not strict and the approaches that are used to achieve this are different depending on the application and the preferences of the manufacturer or the buyer of the device [6]. There are however several characteristics that shape the appearance of the collaborative robot in general. These characteristics show on both the hardware and software side of the system and are partly influenced by safety standards and regulations. Physically, the manipulators often have a relatively lightweight and sleek design.



(a) Collaboration between a robot and its operator[1]

(b) High performance robot manipulators working in an enclosed perimeter[2]

Figure 1-1: Comparison of non-collaborative and collaborative robot manipulators. The manipulator in (a) shows compliant behavior and allows for the operator to be present during operation. Welding robots, as shown in (b), do not feature compliant behavior. Operators are not allowed near the systems during operation.

Their joints may contain springs or other flexible mechanisms to increase the compliance of the device. A high number of joints makes them dexterous and versatile, so they can execute a great number of tasks. In most cases, additional sensors are implemented, such as force/torque sensors, to improve safety and control capabilities. The interface is relatively user-friendly and software that is used to control the robot is generally advanced, but easily programmable (for example through teaching by demonstration) [6].

The application of collaborative robots varies considerably compared to their non-collaborative counterparts. Traditionally, industrial robots are used in various areas where they are deployed to do jobs independently of an operator. These jobs include material handling, packaging, welding and other high-precision operations, as well as jobs that can be described as dirty, dull or dangerous [7]. In many cases these robots are programmed to do one task, and one task only. A good example of such a robot is a welding robot shown in Figure 1-1b. With advancements in technology, collaborative robots have emerged. Through this, the field in which industrial robots are deployed is expanding and the number of tasks that can be executed is increasing. The typical attributes of collaborative robots that are described previously, make that they are especially qualified to execute tasks to *assist*, rather than to work independently [6]. Consequently, tasks like lifting (heavy) parts during manual assembly or repositioning objects for further use can be executed by such devices. Additionally, due to their compliance and safety instructions, the manipulators can work alongside humans without the necessity of a safety cage. As a result, humans and robots can work on a single object at the same time, leading to increased efficiency [6]. Figure 1-1a shows such collaboration between a robot and its operator.

Given the rise in interest in collaborative devices, numerous robot manufacturers have seized the opportunity to expand their product line-up and design collaborative robots for this emerging market. Some big names in the industry, such as ABB, Fanuc, Kuka, and Bosch, all have collaborative robots in production. Other companies like Rethink Robotics and Universal Robotics develop and build collaborative robots exclusively. Most of these systems

[1] Source: http://www02.abb.com/global/seitp/seitp202.nsf/0/93444951d1557c59c1257e200051d731/$file/YuMi.jpg

[2] Source: http://www.roboticstrends.com/images/wide/roboticarms.jpg

differ in size and capabilities, but show similar characteristics based on the aforementioned description on collaborative robots. A convenient overview of several of such devices is found in [5].

1-2 Research description and motivation

1-2-1 Benefits of accurate system knowledge

Generally, collaborative robots are designed for various tasks, using advanced control strategies. Such tasks include impedance control and gravity compensation. These control strategies rely on a forward control action which is based on a predefined model of the manipulator dynamics. Having an accurate model of the system is vital for the effectiveness of the controller and the manipulator itself. Several valuable positive consequences of accurate model knowledge for control are stated below.

Allows for lower feedback gains

With an accurate feedforward action, the accuracy of the operation relies less on the feedback action. Using higher feedback gains improves accuracy, but also has some downsides such as reduced compliance, risk of actuator saturation and instability in the presence of noise or unmodeled dynamics [8]. Additionally, high feedback gains increase the stiffness which may invoke a more intense reaction in the case of a collision.

Allows for more accurate impedance control

For impedance control, the measured forces are considered in combination with the dynamics model of the manipulator [9]. When the model deviates from the true system, expected and measured forces deviate, resulting in a bias introduced by the controller itself.

Prevents drift during gravity compensation tasks

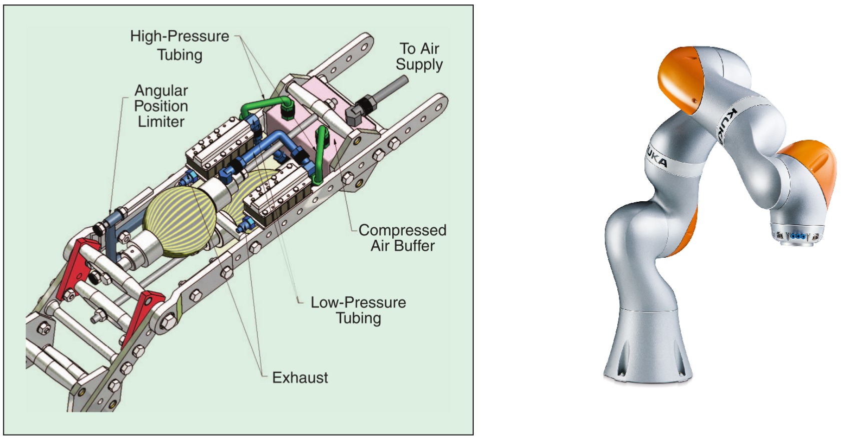
The same holds for gravity compensation [9]. Discrepancies between the internal model and reality cause the controller to interpret these model inaccuracies as external forces. As a consequence, the manipulator will act accordingly, introducing a drifting motion.

Facilitates the detection of external interaction

Analogous to registering non-existing forces is overlooking existing forces. This phenomena may cause damage to the device or harm humans in the vicinity of the device.

1-2-2 Challenges

Motivated by the statements above, acquiring accurate knowledge of the system dynamics has a high priority. However, obtaining accurate system knowledge is challenging under several conditions. Two primary conditions considered in this thesis are described below.



(a) Pneumatic actuator as used in the biped Lucy [11]. Image adopted from [10]

(b) Kuka iiwa LBR 14 R820 collaborative robot[3]

Figure 1-2: Manipulator characteristics that induce challenges for control. (a) shows a configuration that is actuated through pneumatic actuators. Innovative actuators (like these) may introduce modeling errors. (b) shows a lightweight collaborative robot with a payload-to-weight ratio of nearly 50%. Therefore, the influence of the payload on the system dynamics is relatively high. More information on this device is found at [12].

Uncertain manipulator dynamics

The dynamics of a manipulator are described using models which only partly describe reality. This is especially relevant for passively compliant robot manipulators due to their often complex mechanical designs. Examples of mechanical parts that may prove difficult to model accurately are flexible links, passive compliant joints and compliant actuators [10]. An example of such a compliant actuator is shown in Figure 1-2a. Modeling errors may also occur when the system dynamics are changing over time due to varying operating conditions or due to wear of components. Any incorrectly modeled or unmodeled dynamics lead to incorrect control actions.

Significant instantaneous system changes

In addition to unknown manipulator dynamics, the system may also be exposed to (instantaneous) changes to the environment or to the manipulator itself. The most common example of this would be a pick and place task of an unknown object. The inertial parameters of this object must be considered and compensated for by the controller. When the inertial parameters are significant compared to the dynamics of the manipulator itself, this instantaneous system change may severely impact controller performance. With the introduction of *lightweight robots*, payload-to-weight ratios have increased to nearly 50% (The *Kuka LBR iiwa 14 R820*, displayed in Figure 1-2b, has a mass of approximately 30 kg and accepts payloads of 14 kg). In practice, the operator has to find and fix the inertial parameters of the considered object in advance. Alternatively, high-tech devices (such as the Kuka LBR iiwa) feature the

[3] Source: http://www.utkuolcar.com/wp-content/uploads/2017/01/kuka-iiwa.jpg

capability to identify these parameters by maneuvering the object over a predefined set of trajectories.

1-2-3 Problem formulation

From the subsections above it is concluded that ideally, perfect knowledge of the system dynamics is available to guarantee accurate control capabilities and load estimations. However, innovative manipulator designs with potentially flexible links and joints, or passive compliant actuators make modeling the dynamic relations difficult. The introduction of instantaneous system changes, for example through the picking and placing of unknown objects, adds to these complications. In addition, increasing payload-to-weight ratios amplify the significance of unknown objects on the dynamics model mismatch. In order to solve these problems, the following research goal is formulated:

Construct an approach for identifying the dynamics of a robot manipulator which is subjected to both model uncertainties as well as instantaneous system changes.

Existing research is mainly concerned with learning manipulator dynamics under time-invariant conditions. Recent articles with this objective are [13] and [14]. Time-varying conditions are not taken into account here. Other research is focused on modeling and identifying instantaneous system changes. These include multiple-model methods such as described in [15] and [16], or methods that utilize parameter adaptation as described in [7] or [17]. Here however, the assumption is made that the system behaves perfectly according to the rigid body dynamics (RBD) description. This assumption may not hold for the manipulators considered in this thesis. Hence, the available methods do not provide the means to achieve the desired goal.

In order to achieve the stated goal, several objectives are derived. The objectives form the guideline for the remainder of this thesis. The specific case considered in this thesis is based on the pick and place task. A rigid object is assumed to be attached to the end effector of a manipulator, of which the dynamics are uncertain and partly incorrect. Further assumptions and imposed restrictions are discussed in later parts of the report, when relevant.

Specify the manipulator model
The aim of this thesis is to facilitate working with collaborative robot manipulators. To obtain relevant results, an appropriate model must be used to describe these manipulators.

Provide an approach for indentifying/learning the dynamic relations
The behavior of the system must be observed and identified. The proposed method must be able to handle the model uncertainties and system changes that are implied in the research goal.

Investigate the performance of the proposed scheme
The validity of the obtained approach is investigated. This is done based on the results obtained in simulation.

In this thesis, an attempt is made to achieve the research goal making use of both a model based description and machine learning concepts. The combined *semi-parametric* approach ideally captures the advantages of both methods. As such, model knowledge is used to obtain a general description of the system dynamics and machine learning is used to deal with unmodeled dynamics. This attempt is motivated by the argument that “Any application area that uses regression analysis can benefit from semi-parametric regression.” [18]

1-3 Thesis outline

In the remainder of this thesis, the problem stated in Subsection 1-2-3 is further discussed and solutions to tackle the problem are proposed. In Chapter 2, the used modeling methods are described. In Chapter 3, model learning and identification approaches are introduced and discussed. Subsequently, a method for the integration of the proposed algorithms is described in Chapter 4. The proposed identification approach will be evaluated in Chapter 5. Lastly, conclusions and recommendations are stated in Chapter 6.

Chapter 2

Modeling

The first objective of this thesis states that a manipulator model is to be defined. Collaborative robots prove to be difficult to model due to passive compliant actuators or flexible links or joints. Capturing these characteristics in the model would make the used model more complex and not generally applicable. For these reasons it is decided that a generic *rigid body dynamics (RBD) model* is used for the remainder of the thesis. Take note that model mismatch and uncertainties can be introduced regardless of the used modeling method, ensuring that the obtained results are equally valid.

In addition, the following modeling characteristics are assumed:

- The manipulator uses the *articulated layout*, consisting only of revolute joints in a serial configuration
- The system is actuated through *torque control* at every joint
- Instantaneous system changes are represented by modeling a *rigid object with unknown dynamic parameters*, which engages at the end effector of the device.

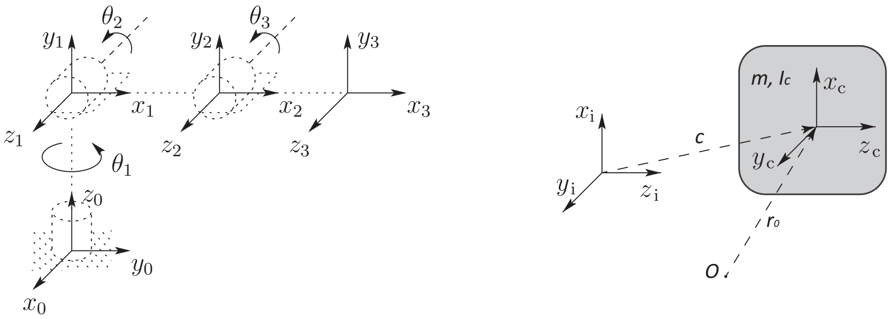
Throughout this thesis, the considered manipulator is described with reference to the *joint space coordinates*. The *system state* is defined by the angular position $\boldsymbol{q}$, angular velocity $\dot{\boldsymbol{q}}$ and angular acceleration $\ddot{\boldsymbol{q}}$. The Cartesian *base frame coordinates* ($\boldsymbol{r}$, $\dot{\boldsymbol{r}}$ and $\ddot{\boldsymbol{r}}$) are relate to the joint space coordinates as:

$$\boldsymbol{r} = g(\boldsymbol{q}) \tag{2-1a}$$

$$\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2-1b}$$

$$\ddot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2-1c}$$

where $\boldsymbol{J}(\boldsymbol{q})$ is referred to as the *Jacobian*. The nonlinear function $g(\boldsymbol{q})$ is defined based on the kinematics of the manipulator. The remainder of this thesis does not discuss the theory and practice of operational space control but focusses on the dynamic relations in the joint-space. In [19], more information on the topic of operational space control is given.



(a) Representation of a serial manipulator. Figure adopted from [7]

(b) Rigid object in a translated reference frame

Figure 2-1: Reference frames for kinematics and dynamics. The reference frames in (a) are located at each joint. Each frame can be represented as a transformation of the previous frame. (b) Shows a rigid object (indicated by the gray area) with its own axes ($\boldsymbol{x}_c$, $\boldsymbol{y}_c$ and $\boldsymbol{z}_c$) in a translated reference frame ($\boldsymbol{x}_i$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$). $\boldsymbol{r}_0$ represents the vector from the base frame origin to the object center of mass.

For the kinematic description of the manipulator, the Denavit-Hartenberg (DH) representation is considered. In this representation, a reference frame is attached to each joint of the manipulator (as shown in Figure 2-1a). Each frame can be derived by a transformation (rotation and translation) of the previous reference frame. A transformation is uniquely defined by four parameters: θ, a, d and α, which are referred to as the *link length*, *link twist*, *link offset* and *joint angle* respectively. A detailed description of the DH representation is found in [7].

The dynamics of a rigid body manipulator are considered next. Various methods are developed for deriving these equations. In this thesis, two methods are considered relevant: the Euler-Lagrange method and the Newton-Euler algorithm [7]. The Newton-Euler algorithm is particularly convenient for efficient numerical computations. The Euler-Lagrange formulation uses an energy-based approach which can be used to obtain a symbolic representation of the manipulator. This chapter continues with Section 2-1, in which a short description is given to the implementation of the Newton-Euler algorithm for a serial manipulator. Subsequently, in Section 2-2, the Euler-Lagrange equations are used to obtain a linear representation of a rigid object in a translated reference frame. Section 2-3 ends this chapter with a discussion.

2-1 Dynamics of a serial manipulator

Commonly, the equations of motion for a robot manipulator are described as $\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$ [7]. With reference to the research goal, this equation can be extended towards:

$$\boldsymbol{\tau}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{M}(t, \boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}(t, \boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \boldsymbol{G}(t, \boldsymbol{q}) + \boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{2-2}$$

The time variable t is added to the equations to indicate the time-varying behavior of the system. The nonlinearities are contained in the matrices $\boldsymbol{M}(t, \boldsymbol{q})$, $\boldsymbol{C}(t, \boldsymbol{q}, \dot{\boldsymbol{q}})$, $\boldsymbol{G}(t, \boldsymbol{q})$ and $\boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, where $\boldsymbol{M}(t, \boldsymbol{q})$ is referred to as the inertia or mass matrix, $\boldsymbol{C}(t, \boldsymbol{q}, \dot{\boldsymbol{q}})$ contains

Table 2-1: Relevant manipulator parameters for RBD

Parameter	**Physical representation**
DH-parameters	Used to describe manipulator kinematics
m	Link mass
$\boldsymbol{c}$	link center of mass
$\boldsymbol{I}$	link inertia matrix around center of mass
J_m	Motor inertia
R	Transmission ratio
B	Viscous friction
T_c	Coulomb friction

the Coriolis and centripetal forces, $\boldsymbol{G}(t, \boldsymbol{q})$ contains gravity forces and $\boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ represents (non)linear, (non-)conservative torques, which are introduced by incorrectly modeled or unmodeled dynamics. The contribution of this term to the complete equation increases when the true manipulator is not accurately described using the RBD model.

When no symbolic representation of the system is needed and fast computation time is required during simulation, often the Newton-Euler algorithm is used. This method applies a forward recursion which sequentially derives the linear and angular motion for each link, from the manipulator base frame to the end effector of the robot (in Figure 2-1a, from the base, frame 0, to the end effector, frame 3). After this forward recursion, a backward recursion is executed where the reaction forces and coupling terms are considered (from frame 3 back to frame 0). Additional torques and nonlinear frictions can be applied directly to the joints during the backward recursion.

In order to apply the Newton-Euler algorithm, full knowledge of the kinematic and dynamic parameters of the manipulator is required. Table 2-1 gives an overview of the relevant parameters.

2-2 Dynamics of a rigid object

As stated in the introduction, one cause for the occurrence of instantaneous system changes is the picking and placing of unknown objects. In this section, the dynamic relations of a rigid object attached to the end effector of a manipulator are described using a symbolic formulation. The behavior of any rigid object in an arbitrary reference frame can be represented using its mass, center of mass and its moment of inertia around this center of mass. Symbolic expressions of these parameters are as displayed in Equations (2-3a) to (2-3c). The center of mass is taken relative to the reference frame and the moment of inertia terms are relative to the object's own center of mass. Ten unique values are required for a full definition.

$$\text{Mass} \quad m = \begin{bmatrix} m \end{bmatrix} \tag{2-3a}$$

$$\text{Center of mass} \quad \boldsymbol{c} = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \tag{2-3b}$$

$$\text{Moment of inertia} \quad \boldsymbol{I}_c = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{2-3c}$$

With relevance to the goals in this thesis, the dynamic relations are desired of a static object in a moving reference frame, which refers to a rigid object being picked up by the manipulator. A visual representation of this is displayed in Figure 2-1b. To describe the dynamic relations, first the kinetic energy $\mathcal{K}$ and potential energy $\mathcal{P}$ of the object are determined. The *kinetic energy* of the object is defined in Equation (2-4a) and is the sum of the linear and angular kinetic energies. Here, variables $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are respectively the linear and angular velocities, where subscript c refers to the center of mass of the object and subscript i refers to the reference frame. The identity matrix is indicated as $\mathbb{I}$. Using $\boldsymbol{v}_c = \boldsymbol{v}_i + \boldsymbol{\omega}_i \times \boldsymbol{c}$ and the parallel axis theorem: $\boldsymbol{I}_c = \boldsymbol{I}_i - m\left(\boldsymbol{c}\boldsymbol{c}^\intercal \mathbb{I}_{3\times 3} - \boldsymbol{c}^\intercal \boldsymbol{c}\right)$, the kinetic energy can be rewritten into Equation (2-4b). Transformations indicated by $S_\times$ and S_I (which are defined in Equations (A-1) and (A-2)) are used to obtain the linear parameter representation in Equation (2-4c) [20].

$$\begin{aligned} \mathcal{K} &= \frac{1}{2} m \boldsymbol{v}_c^\intercal \boldsymbol{v}_c + \frac{1}{2} \boldsymbol{\omega}_i \boldsymbol{I}_c \boldsymbol{\omega}_i && \text{(2-4a)} \\ &= \frac{1}{2} m \boldsymbol{v}_i^\intercal \boldsymbol{v}_i + m \boldsymbol{c}^\intercal S_\times(\boldsymbol{v}_i) \boldsymbol{\omega}_i + \frac{1}{2} \boldsymbol{\omega}_i^\intercal S_I(\boldsymbol{\omega}_i) \boldsymbol{I}_i && \text{(2-4b)} \\ &= \underbrace{\begin{bmatrix} \frac{1}{2} \boldsymbol{v}_i^\intercal \boldsymbol{v}_i, & \boldsymbol{\omega}_i^\intercal S_\times^\intercal(\boldsymbol{v}_i), & \frac{1}{2} \boldsymbol{\omega}_i^\intercal S_I(\boldsymbol{\omega}_i) \end{bmatrix}}_{\mathcal{K}_R(\boldsymbol{q}, \dot{\boldsymbol{q}})} \underbrace{\begin{bmatrix} m \\ m\boldsymbol{c} \\ \boldsymbol{I}_i \end{bmatrix}}_{\boldsymbol{w}} && \text{(2-4c)} \end{aligned}$$

The *potential energy* is composed in a similar fashion, where the linear parameters are extracted from Equation (2-5a) to obtain the linear representation in Equation (2-5c). Parameter $\boldsymbol{r}_0$ refers to the vector from the base frame origin of the manipulator to the end effector joint. $\boldsymbol{g}_0$ is the gravity vector, defined as: $\boldsymbol{g}_0 = [0,\ 0,\ 9.81]^\intercal$.

$$\begin{aligned} \mathcal{P} &= -m \boldsymbol{g}_0 \left(\boldsymbol{r}_0 + \boldsymbol{c}\right) && \text{(2-5a)} \\ &= -m \boldsymbol{g}_0^\intercal \boldsymbol{r}_0 - \boldsymbol{g}_0^\intercal \left(m\boldsymbol{c}\right) && \text{(2-5b)} \\ &= \underbrace{\begin{bmatrix} -\boldsymbol{g}_0^\intercal \boldsymbol{r}_0, & -\boldsymbol{g}_0^\intercal, & \boldsymbol{0} \end{bmatrix}}_{\mathcal{P}_R(\boldsymbol{q})} \underbrace{\begin{bmatrix} m \\ m\boldsymbol{c} \\ \boldsymbol{I}_i \end{bmatrix}}_{\boldsymbol{w}} && \text{(2-5c)} \end{aligned}$$

The equations of motion for the considered object are now derived using the Euler-Lagrange description in Equation (2-6a). In Equation (2-6b), the product notation introduced to the kinetic and potential energy is maintained. The dynamics are now written as a product of a state-dependent *regressor* and an independent *linear parameter vector*: $\boldsymbol{\tau}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = \boldsymbol{\Phi}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\,\boldsymbol{w}$.

$$F_{\text{ext}} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{K}}{\partial \dot{\boldsymbol{q}}} - \frac{\partial \mathcal{K}}{\partial \boldsymbol{q}} + \frac{\partial \mathcal{P}}{\partial \boldsymbol{q}} \tag{2-6a}$$

$$= \underbrace{\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{K}_R^{\intercal}}{\partial \dot{\boldsymbol{q}}} - \frac{\partial \mathcal{K}_R^{\intercal}}{\partial \boldsymbol{q}} + \frac{\partial \mathcal{P}_R^{\intercal}}{\partial \boldsymbol{q}} \end{bmatrix}^{\intercal}}_{\boldsymbol{\Phi}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})} \underbrace{\begin{bmatrix} m \\ m\boldsymbol{c} \\ \boldsymbol{I}_i \end{bmatrix}}_{\boldsymbol{w}} \tag{2-6b}$$

2-3 Discussion

This chapter describes how the manipulator and a (to the end effector attached) rigid object are described using a RBD model. A number of decisions are made to specify the complete system used for this thesis. This section is used to demonstrate that these assumptions do not decrease the relevance of the research.

2-3-1 The relevance of the RBD model

The RBD model is the primary parametric method used for modeling the dynamics of the manipulator in this thesis. The motivation to use the RBD model is its simplicity and interpretability. Considering the statements made in the introduction, this type of model may prove to be not entirely valid for the discussed manipulators.

Taking this issues into account, the RBD model is utilized nevertheless. The justification for this decision is based on the potential to make adaptations to the parameters during operation and the possibility to add arbitrary torques to each joint. Consequently, nonlinear relations which are in reality introduced by parameter uncertainties, time-varying phenomena, nonlinear joint frictions and unknown actuator characteristics, may be introduced to the model regardless of the RBD model used initially.

2-3-2 Manipulator control

The manipulator assumed for this thesis uses the articulated layout and is controlled by applying toques directly to each joint. The assumption of torque control is convenient for describing the system dynamics and simulating the robot. The potential to use torque control on industrial manipulators is somewhat limited for end users. However, the assumption of torque control implies control on a low level, which increases the validity of the proposed method in general. This thesis is concerned with the proof of concept of the constructed approach. Therefore, torque control remains a valid solution. Additionally, the assumed low-level control provides the potential to include actuator modeling errors. The modeling errors directly influence the torques applied to the joints. Torque control therefore gives the most fitting and practical representation for this thesis.

Chapter 3

Identification of system dynamics

The next stage of the report is focused on completing the second objective, which is establishing the approach used for identifying or learning the dynamics of the system. The modeling method used for this purpose is defined in Chapter 2. Even though the model is defined, the system parameters are assumed to be uncertain, which is in accordance with the research goal. Furthermore, instantaneous system changes are also to be taken into account.

With respect to identification and learning, the following *complementary statements* are made:

- Identification is motivated by improving the validity of the system its dynamics equations (see Equation (2-2)) for control purposes, not to obtain the true physical representation of the system.
- To prevent down-time, an approach is required that allows for identification during operation
- Adaptation to system changes must occur in an automated fashion

In [21], a survey is found which is focused on modeling, identification and machine learning. Various characteristics are used to describe the methods concerned with this subject. The distinction is made between parameteric and non-parametric methods, global and local methods and also online and offline methods. In addition, differences in the complexity of these methods may also be taken into account.

The aim in this thesis is to create an advantage by using prior knowledge. To this end, the rigid body dynamics (RBD) are used to describe the input-output relations of the manipulator. This *parametric method* gives a globally valid model when all dynamic system parameters are known. These parameters may be provided by the manufacturer of the manipulator. The assumed rigid object can be described using RBD as well. For the purpose of this thesis, the relevant parameters of this object are assumed to be unknown. Given a set of input and output data, the unknown parameters may in this case be identified using least squares algorithms [22]. Alternatively, stochastic methods such as the Kalman filter can be used to identify unknown parameters [17].

The dynamics of the system can also be described through the use of *non-parametric algorithms*. These types of algorithms do not use a predefined structure and are in this sense more flexible, allowing for a better fit. Several of such methods are developed to solve the identification problem through regression. Some popular algorithms include *support vector regression* [23], *locally weighted projection regression* [24] and *Gaussian process regression (GPR)* [25]. In this thesis, GPR is considered, by virtue of its high modeling accuracy and interpretability of predictions. Furthermore, GPR allows for *semi-parametric regression*, which is also proposed in [26] and because of its stochastic nature, the likelihood of measured data can be determined. The RBD are used as prior knowledge of the GPR algorithm, ensuring that acceptable results are found when only a limited amount of data is available.

The remainder of this chapter further describes the mentioned frameworks. Section 3-1 continues on how the extended Kalman filter (EKF) can be used to identify unknown parameters. In Section 3-2, an introduction is given to GPR, followed by a description of how it is used to describe manipulator dynamics. Finally, the last section (Section 3-3) of this chapter contains a discussion on these frameworks.

3-1 Temporal identification using the extended Kalman filter

A widely accepted algorithm for state estimation is the Kalman filter. The Kalman filter may also be applied to observe unknown system parameters. The equations on which the Kalman filter problem is based read:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{\mu}_{k-1} \tag{3-1a}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{\nu}_k, \tag{3-1b}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are respectively the process state and output at time step k. The function $f(\cdot)$ relates the previous state to the current state and the function $h(\cdot)$ relates the current state to the output. The state and output equations are subjected to noise terms $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, which are modeled as zero mean Gaussian processes.

3-1-1 Technical description of the EKF

The considered system for this thesis is nonlinear, so the conventional linear Kalman filter does not suffice. Consequently, a first order EKF is proposed for this research, because it has proven itself useful in similar cases [17] and requires much less computational effort than alternative methods, such as the *unscented Kalman filter* [27]. The EKF is derived from the linear Kalman filter, but uses the Jacobians presented in Equations (3-2a) and (3-2b) for the prediction of the covariance $\boldsymbol{P}$ of the state $\boldsymbol{x}$ and the calculation of the Kalman gain $\boldsymbol{K}$. Matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ represent the covariance matrices of the state equation noise and measurement equation noise respectively. The recursive equations for the EKF are displayed in Equations (3-3a) to (3-3f), where the predicted state and state covariance matrix (before the update) are indicated by $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{P}}$.

$$\boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{x}_{k-1}) = \left.\frac{\partial f(\boldsymbol{x}_i)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_i=\boldsymbol{x}_{k-1}} \tag{3-2a}$$

$$\boldsymbol{H}_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_k) = \left.\frac{\partial h(\tilde{\boldsymbol{x}}_i)}{\partial \boldsymbol{x}}\right|_{\tilde{\boldsymbol{x}}_i=\tilde{\boldsymbol{x}}_k} \tag{3-2b}$$

$$\text{prediction:} \quad \tilde{\boldsymbol{x}}_k = f(\boldsymbol{x}_{k-1}) \tag{3-3a}$$

$$\tilde{\boldsymbol{P}}_k = \boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{x}_{k-1})\boldsymbol{P}_{k-1}\boldsymbol{F}_{\boldsymbol{x}}^{\intercal}(\boldsymbol{x}_{k-1}) + \boldsymbol{M}_{k-1} \tag{3-3b}$$

$$\text{update:} \quad \boldsymbol{S}_k = \boldsymbol{H}_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_k)\tilde{\boldsymbol{P}}_k\boldsymbol{H}_{\boldsymbol{x}}^{\intercal}(\tilde{\boldsymbol{x}}_k) + \boldsymbol{N}_k \tag{3-3c}$$

$$\boldsymbol{K}_k = \tilde{\boldsymbol{P}}_k\boldsymbol{H}_k^{\intercal}(\tilde{\boldsymbol{x}}_k)\boldsymbol{S}_k^{-1} \tag{3-3d}$$

$$\boldsymbol{x}_k = \tilde{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - h(\tilde{\boldsymbol{x}}_k)\right) \tag{3-3e}$$

$$\boldsymbol{P}_k = \tilde{\boldsymbol{P}}_k - \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^{\intercal} \tag{3-3f}$$

3-1-2 Implementation of the EKF for parameter identification

As is described in Section 2-2, the dynamic equations of an object that is attached to the end effector of a manipulator can be expressed as: $\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\boldsymbol{w}$. To identify the unknown parameters $\boldsymbol{w}$, the EKF is used, where the linear parameter vector is defined as the state. To prevent unrealistic estimates, boundary conditions on the state vector are applied. This is achieved by transforming the state through a *sigmoid function*, as proposed in [17]. For the transformations, see Equations (A-3) to (A-5). The prediction and update equations with the transformed state $\boldsymbol{w}' = \mathrm{Sig}^{-1}(\boldsymbol{w})$ are written as in Equations (3-4a) and (3-4b). The state is assumed constant, with the Jacobian as in Equation (3-5a). The Jacobian of the measurement update is given in Equation (3-5b).

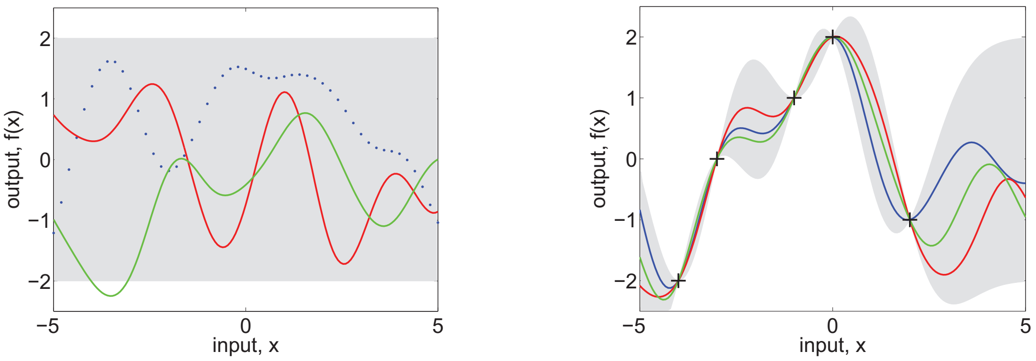
$$\boldsymbol{w}'_k = \boldsymbol{w}'_{k-1} + \boldsymbol{\mu}_{k-1} \tag{3-4a}$$

$$\boldsymbol{\tau}_k = \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\mathrm{Sig}(\boldsymbol{w}'_k) + \boldsymbol{\nu}_k \tag{3-4b}$$

$$\boldsymbol{F}_{\boldsymbol{w}'}(\boldsymbol{w}'_{k-1}) = \left.\frac{\partial f(\boldsymbol{w}'_i)}{\partial \boldsymbol{w}'}\right|_{\boldsymbol{w}'_i=\boldsymbol{w}'_{k-1}} = \mathbb{I}_{10\times 10} \tag{3-5a}$$

$$\boldsymbol{H}_{\boldsymbol{w}'}(\tilde{\boldsymbol{w}}'_k) = \left.\frac{\partial h(\tilde{\boldsymbol{w}}'_i)}{\partial \boldsymbol{w}'}\right|_{\tilde{\boldsymbol{w}}'_i=\tilde{\boldsymbol{w}}'_k} = \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\left.\frac{\partial\,\mathrm{Sig}(\tilde{\boldsymbol{w}}'_i)}{\partial \boldsymbol{w}'}\right|_{\tilde{\boldsymbol{w}}'_i=\tilde{\boldsymbol{w}}'_k} \tag{3-5b}$$

The used sigmoid function has an additional benefit, which is described in [17]. The derivative of the sigmoid function goes to zero when the upper or lower bound is reached. Consequently, the Jacobian in Equation (3-5b) decreases for the states that approach their imposed bounds. This effect slows down the rate at which that specific state is allowed to change.



(a) Prior distribution

(b) Posterior distribution

Figure 3-1: Functions drawn at random from a Gaussian distribution. The Gaussian distribution has a zero mean and the marked grey area indicates plus and minus two times the standard deviation (a). When the GP is conditioned on the training data (marked with the '+'-symbols), a posterior distribution is found (b). The figures are adopted from [25].

3-2 Spatial description using Gaussian process regression

Even though identifying the dynamics of nonlinear systems is feasible using strictly parametric descriptions, inaccuracies occur when the assumed model structure is incorrect or incomplete. In this section, the Gaussian process (GP) framework is introduced, which is described in detail in [25]. This framework can be utilized to solve regression problems. Through regression an input-output relation is learned after which output predictions can be made based on a given input.

3-2-1 Regression in the GP framework

The goal of a regression method is to enable the prediction of a continuous variables. The GP is the tool which is used to achieve this goal. In the GP framework, inference often takes place in the *function space*. The non-parametric nature ensures a high flexibility and the possibility to model nonlinear functions [25].

The stochastic variables used for the GPR framework may be defined through any kind of *probability density function*, however most often (as is the case in this thesis), the variables are assumed to satisfy a Gaussian distribution. The multivariate Gaussian distribution is defined as:

$$\mathcal{N}\left(\boldsymbol{\mu}\,,\,\boldsymbol{\Sigma}\right) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right), \tag{3-6}$$

and is completely defined by two parameters: the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$.

In general, the goal of a regression method is to find the latent function $f(\boldsymbol{X})$ of the stochastic process from Equation (3-7), using the n aggregated inputs $\boldsymbol{x} \in \mathbb{R}^{1\times m}$ in matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ and outputs in vector $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$. The latent function $f(\boldsymbol{X})$ may be represented by any linear or nonlinear function and the term $\boldsymbol{\epsilon} \in \mathbb{R}^{n\times 1}$ is often assumed to be a Gaussian distributed white noise signal with a variance of σ_n^2, i.e.: $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}\,,\,\sigma_n^2\mathbb{I}_{n\times n}\right)$.

$$\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon} \tag{3-7}$$

The probability distribution of the latent function values for a given set of inputs and outputs is called the *posterior distribution* and is found by applying *Bayes' rule* as indicated in Equation (3-8). In the GP framework, not only the noise but also the latent function itself is modeled as a GP, defined over the input space. Consequently, the assumption can be made that the latent function $f(\boldsymbol{X})$ behaves according to a zero mean GP as displayed in Equation (3-9a). This distribution is often referred to as the *prior*, since this term contains prior assumptions of the latent function, which is gathered in the covariance matrix $\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}$ and sometimes also the mean function. The used notation is: $\boldsymbol{K}_{\boldsymbol{a},\boldsymbol{b}} \in \mathbb{R}^{n1\times n2} = \text{cov}(\boldsymbol{a} \in \mathbb{R}^{n1\times m}, \boldsymbol{b} \in \mathbb{R}^{n2\times m})$, where the function $\text{cov}(\cdot\,,\cdot)$ represents the used covariance function. With GPR, the covariance matrix is often constructed based on the *squared exponential covariance function*. More information on this subject can be found in [25]. A visual representation of the prior is displayed in Figure 3-1a. Continuing on Equation (3-7) and with the noise defined as $\boldsymbol{\epsilon} = \mathcal{N}\left(\boldsymbol{0}\,,\,\sigma_n^2\mathbb{I}\right)$, the probability distribution of the output values, given the latent function, is referred to as the *likelihood* and reads as in Equation (3-9b). The term in the denominator of Equation (3-8) represents the *marginal likelihood* and is a normalizing constant which is independent of the latent function values.

$$\text{posterior} = p\left(f(\boldsymbol{X})\,|\,\boldsymbol{y}\right) = \frac{p\left(\boldsymbol{y}\,|\,f(\boldsymbol{X})\right)p(f(\boldsymbol{X}))}{p(\boldsymbol{y})} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \tag{3-8}$$

$$p(f(\boldsymbol{X})) = \mathcal{N}\Big(\boldsymbol{0}\,,\,\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}\Big) \tag{3-9a}$$

$$p\left(\boldsymbol{y}\,|\,f(\boldsymbol{X})\right) = \mathcal{N}\Big(f(\boldsymbol{X})\,,\,\sigma_n^2\mathbb{I}\Big) \tag{3-9b}$$

$$p(\boldsymbol{y}) = \int p(f(\boldsymbol{X}))\,p\left(\boldsymbol{y}\,|\,f(\boldsymbol{X})\right)\mathrm{d}\boldsymbol{X} \tag{3-9c}$$

The posterior distribution is proportional to the product of the likelihood and the prior. Since the prior and likelihood both have a Gaussian distribution, the posterior distribution also satisfies a Gaussian distribution and is represented by the statement in Equation (3-10). The latent function values $f(\boldsymbol{X})$ are indicated as $\boldsymbol{f}_{\boldsymbol{x}}$. For a Gaussian posterior, the mean is also the *maximum a posteriory*, implying that with the assumed noise model and inputs $\boldsymbol{X}$, the probability of measuring outputs $\boldsymbol{y}$ from latent function values $f(\boldsymbol{X}) = \boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I})^{-1}\boldsymbol{y}$ is maximized.

$$\begin{aligned} p\left(\boldsymbol{f}_{\boldsymbol{x}}\,|\,\boldsymbol{y}\right) \propto p\left(\boldsymbol{y}\,|\,\boldsymbol{f}_{\boldsymbol{x}}\right)p(\boldsymbol{f}_{\boldsymbol{x}}) &= \mathcal{N}\Big(\boldsymbol{f}_{\boldsymbol{x}}\,,\,\sigma_n^2\mathbb{I}\Big)\,\mathcal{N}\Big(\boldsymbol{0}\,,\,\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}\Big) \\ &= \mathcal{N}\Big(\sigma_n^{-2}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}^{-1} + \sigma_n^{-2}\mathbb{I})^{-1}\boldsymbol{y}\,,\,(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}^{-1} + \sigma_n^{-2}\mathbb{I})^{-1}\Big) \\ &= \mathcal{N}\Big(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I})^{-1}\boldsymbol{y}\,,\,\sigma_n^2\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I})^{-1}\Big) \end{aligned} \tag{3-10}$$

To predict the function values corresponding to an arbitrary input, note that the latent function values corresponding to both training data $\boldsymbol{X}$ and test data $\boldsymbol{X}_*$ are both drawn from a joint Gaussian distribution. Taking into account that the latent function values are only available through their noisy measurements, the joint distribution can be displayed as in Equation (3-11). The conditional probability of the predicted function values $\boldsymbol{f}_*$, given the measured outputs of the training data, is shown in Equation (3-12). Figure 3-1b gives a graphical representation of a posterior distribution.

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}\,,\, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I} & \boldsymbol{K}_{\boldsymbol{x},*} \\ \boldsymbol{K}_{*,\boldsymbol{x}} & \boldsymbol{K}_{*,*} \end{bmatrix}\right) \tag{3-11}$$

$$\boldsymbol{f}_* \sim \mathcal{N}\left(\boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I}\right)^{-1}\boldsymbol{y}\,,\, \boldsymbol{K}_{*,*} - \boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I}\right)^{-1}\boldsymbol{K}_{\boldsymbol{x},*}\right) \tag{3-12}$$

In this subsection the basic description of GPR is given. Additional relevant topics included *covariance functions* ans *hyperparameter optimization*, which are also described in [25].

3-2-2 Modeling inverse dynamics with GPR

The described GPR algorithm can be applied for manipulator control. This is done by training the GP to represent the inverse dynamics of the manipulator [28]. Assuming torque control, the equation described by the GP will be a *spatial mapping* of the form: $\boldsymbol{\tau} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. With reference to a trained data set consisting of inputs ($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$) and outputs ($\boldsymbol{\tau}$), the posterior distribution is found. Using the notation from Subsection 3-2-1, the training inputs are stored in a data matrix: $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, where the rows are composed of n sampled input vectors $[\boldsymbol{q}_i,\, \dot{\boldsymbol{q}}_i,\, \ddot{\boldsymbol{q}}_i] \in \mathbb{R}^{1\times m}$. For a certain array of test inputs $\boldsymbol{X}_*$, the predicted torques are determined to be the mean of the posterior distribution, $\bar{\boldsymbol{\tau}}_*$.

More specifically, using the prior $\boldsymbol{\tau}(\boldsymbol{X}) \sim \mathcal{N}\left(f_{\text{est}}(\boldsymbol{X})\,,\, \boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}\right)$ and the likelihood $\boldsymbol{\tau} \,|\, \boldsymbol{\tau}(\boldsymbol{X}) \sim \mathcal{N}\left(\boldsymbol{\tau}(\boldsymbol{X})\,,\, \sigma_n^2\mathbb{I}\right)$, the predicted torques are computed as:

$$\bar{\boldsymbol{\tau}}_* = f_{\text{est}}(\boldsymbol{X}_*) + \boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I}\right)^{-1}(\boldsymbol{\tau} - f_{\text{est}}(\boldsymbol{X}_*)) \tag{3-13}$$

and its covariance as:

$$\text{cov}(\boldsymbol{\tau}_*) = \boldsymbol{K}_{*,*} - \boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I}\right)^{-1}\boldsymbol{K}_{\boldsymbol{x},*} \tag{3-14}$$

The above statements imply that an arbitrary mean function ($f_{\text{est}}(\boldsymbol{X})$) is used for the prior distribution. Using a non-zero mean is beneficial due to higher modeling accuracy, better generalization and a faster learning speed [26]. In the absence of relevant data, the second term of Equation (3-13) drops to (near) zero and prediction will read: $\bar{\boldsymbol{\tau}}_* \approx f_{\text{est}}(\boldsymbol{X}_*)$.

When the notation $f_{\text{est}}(\boldsymbol{X}) = \boldsymbol{\Phi}(\boldsymbol{X})\,\boldsymbol{w}$ holds, the parameter vector may be assumed to be a stochastic variable defined as $\boldsymbol{w} = \mathcal{N}\left(\bar{\boldsymbol{w}}\,,\, \boldsymbol{B}\right)$, where $\boldsymbol{B}$ is a covariance matrix that contains uncertainty measures of the linear parameters in $\boldsymbol{w}$ [25, 26]. Using this, the uncertainties of the linear parameters are accounted for. Based on the training data, the estimated parameter vector is then determined as:

$$\bar{\boldsymbol{w}} = \left(\boldsymbol{B}^{-1} + \boldsymbol{\Phi}(\boldsymbol{X})^{\intercal}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I})^{-1}\boldsymbol{\Phi}(\boldsymbol{X})\right)^{-1}\left(\boldsymbol{\Phi}(\boldsymbol{X})^{\intercal}(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I})^{-1}\boldsymbol{\tau} + \boldsymbol{B}^{-1}\boldsymbol{w}\right) \tag{3-15}$$

And is used for prediction as:

$$\boldsymbol{\tau}_* = \boldsymbol{\Phi}(\boldsymbol{X}_*)\bar{\boldsymbol{w}} + \boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2\mathbb{I}\right)^{-1}(\boldsymbol{\tau} - \boldsymbol{\Phi}(\boldsymbol{X})\bar{\boldsymbol{w}}) \tag{3-16}$$

Although this approach with stochastic parameters shows promising results according to [26], the possibility for the implementation in the considered setup is not certain yet, due to the assumption of a time-varying environment.

3-3 Discussion

This chapter describes the two important frameworks that are used in this thesis to identify and learn the dynamics at play. In this section, some additional remarks are made on the assumptions and characteristics of these two methods.

An assumption that is vital for both methods is that the applied torques, as well as the angular positions, angular velocities and angular accelerations are known. This assumption is valid for modern robots where position sensors are accurate enough to allow for differentiation of the position measurement signal. Remaining noise is dealt with through the term $\boldsymbol{\nu}_k$ in the Kalman filter equations. The GP also assumes the presence of noise. Hence, having only access to noisy velocities and accelerations is not a limiting factor.

Furthermore, the EKF and GP models both assume Gaussian noise models. However, both methods also describe a nonlinear relation of the form $\boldsymbol{\tau} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, for which this assumption is generally not valid. For this thesis it is assumed that the considered nonlinear mapping does not introduce critical issues for the EKF and GP approach based on documented experiments with similar applications [17, 26].

The GPR framework has the desirable characteristic of being able to model a wide variety of functions accurately and is also relatively predictable as compared to other non-parametric frameworks [21, 25]. Applying GPR however also entails some difficulties. With regard to the performance of the predictions, sufficient training data in the considered part of the state-space is required since the approach does not generalize towards unknown parts of the state-space. Furthermore, overfitting or underfitting may occur when the GP is not tuned correctly. the tuning in itself is not a trivial task. Information regarding the tuning of GPs is found in [25]. With respect to the practice of implementing the GP, the management of data is to be considered. This is especially true for real-time applications. Whereas prediction has a computational cost of $\mathcal{O}(n)$, training of the GP scales with the cube of the amount of training data, resulting in a cost of $\mathcal{O}(n^3)$ [25]. For this thesis, GPR is considered nevertheless, based on its potential to use in a semi-parametric framework and its stochastic characteristics. Issues regarding the implementation of GPR are the subject of many existing researches ([21] also refers to research done in this area).

Chapter 4

A semi-parametric approach to identification and control

As stated in Chapter 2, the dynamics of the considered system can be described as shown in Equation (2-2). Finding this relation is the main goal of the research. From a more practical point of view, having perfect knowledge of these relations makes it possible to describe a followed trajectory (e.g. through the joint acceleration $\ddot{\boldsymbol{q}}(t)$) as a function of the applied torques and initial conditions. Alternatively, the torques at the joints may be derived as a function of the followed trajectory. These relations are referred to as the *forward dynamics* and *inverse dynamics* respectively and read as:

$$\begin{aligned} \text{Forward dynamics:} \quad & \ddot{\boldsymbol{q}}(t) = f_{\mathrm{f}}(\boldsymbol{\tau}(t), \boldsymbol{q}_0, \dot{\boldsymbol{q}}_0) && \text{(4-1a)} \\ \text{Inverse dynamics:} \quad & \boldsymbol{\tau}(t) = f_{\mathrm{i}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)) && \text{(4-1b)} \end{aligned}$$

For control purposes, especially the inverse dynamics are of interest. With perfect knowledge of the system and the desired system state $[\boldsymbol{q}_d(t), \dot{\boldsymbol{q}}_d(t), \ddot{\boldsymbol{q}}_d(t)]$, the inverse dynamics can be used to derive a torque that is used as a control input and produces exactly this desired state (provided that the angular position and velocity are initialized correctly). This phenomenon is illustrated in Figure 4-1. Motivated by this description, the identification of the system is primarily aimed at finding the inverse dynamics.

For convenience, the research goal of this thesis is restated here: “Construct an approach for identifying the dynamics of a robot manipulator which is subjected to both model uncertainties as well as instantaneous system changes.”

In accordance with this research goal, the used procedure for identification must be carried out during operation to deal with time-varying phenomena. Consequently, the observed dynamics can be used directly to improve the inverse dynamics model, which in turn provides the forward control term. Three criteria are introduced that give an indication on the effectiveness of the used approach to learn the inverse dynamics. Particularly, the inverse dynamics model must:

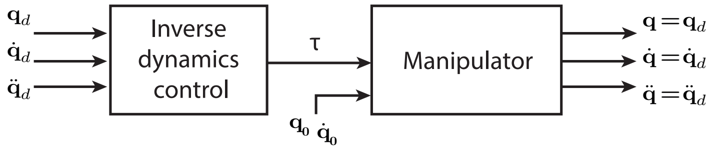

$\mathbf{q}_d$
$\dot{\mathbf{q}}_d$
$\ddot{\mathbf{q}}_d$
Inverse dynamics control
τ
Manipulator
$\mathbf{q}_0$ $\dot{\mathbf{q}}_0$
$\mathbf{q}=\mathbf{q}_d$
$\dot{\mathbf{q}}=\dot{\mathbf{q}}_d$
$\ddot{\mathbf{q}}=\ddot{\mathbf{q}}_d$

Figure 4-1: Inverse dynamics control used to cancel the manipulator dynamics. Given a desired trajectory (existing of $\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{q}}_d$), the torques are derived by the controller and applied to the manipulator. In the case that the inverse dynamics are perfectly known and the initial conditions $\boldsymbol{q}_0$ and $\dot{\boldsymbol{q}}_0$ agree with the desired trajectory, the resulting trajectory exactly resembles the desired trajectory.

- Describe the dynamics accurately.
- Adapt rapidly to deal with instantaneous system changes.
- Give a description that generalizes well over the whole state-space.

The philosophy used in this thesis to meet these criteria is that an attempt should be made to benefit from model knowledge in combination with online learning. This is realized using a *semi-parametric* description of the system dynamics. The semi-parametric approach improves on methods that are purely non-parametric, since the latter do not recognize any structure in the data and therefore do not generalize towards unknown parts of the state-space. Additionally, better modeling accuracy can be achieved compared to strictly parametric methods [26].

In conformity with Chapter 2, a manipulator model is considered with uncertain dynamic relations. Furthermore, the instantaneous model changes are represented by an object which may or may not be attached to the end-effector. Having knowledge of the manner in which a system change may occur allows for the possibility to rapidly identify the transformation and react accordingly. In this thesis, identifying these structured system changes is to be achieved by the extended Kalman filter (EKF). Unstructured and unmodeled errors are dealt with through Gaussian process regression (GPR).

The remainder of the chapter is focused on describing and motivating the semi-parametric approach that is proposed to achieve the research goal stated in the introduction to this thesis. To this end, a control scheme is constructed in Section 4-1, which supports the proposed methods. Information related to observing and identifying during operation is discussed in Section 4-2. Subsequently, the control approach is described in Section 4-3. With this control scheme in place, Section 4-4 continues on system change detection and switching for the designed control configuration. The chapter is ended in Section 4-5 with a discussion and remarks on the treated subjects.

4-1 Control scheme design

In this thesis, identification and learning of the system dynamics is achieved in the context of a control scheme. More specifically, the system is identified based on input and output data that is gathered during operation. The *direct learning* approach is applied to this end (see [21]). Gathered data contributes to the knowledge of the inverse dynamics which can directly

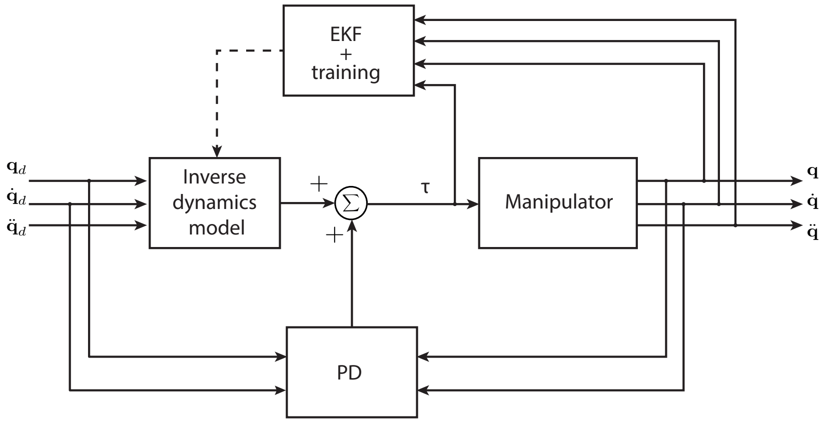


Figure 4-2: Proposed control and identification scheme. Direct learning is implemented to find the inverse dynamics model of the manipulator, which is used for deriving a forward control term. A feedback loop is added to guarantee stability. The control scheme is based on the *feedforward nonlinear control scheme* described in [28].

be utilized to produce an increasingly 'accurate' forward control action, implying that the desired output and true output become more similar. This corresponds with the philosophy represented in Figure 4-1. Since perfect knowledge of the system cannot be assumed, a feedback term is applied to counteract disturbances and deviations from a desired set-points. The torques applied through feedback have a direct influence on the gathered training data and therefore help shape the non-parametric description of the system dynamics. A representation of the identification and control configuration is shown in Figure 4-2. The control term for this configuration reads:

$$\boldsymbol{\tau}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}) = f_{\mathrm{ff}}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) + f_{\mathrm{fb}}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{4-2}$$

where the subscripts *ff* and *fb* refer to the feedforward and feedback terms of which the control action is comprised. subscript d is used to indicate the desired value for a variable. The resulting control torques $\boldsymbol{\tau}$ are applied to the actuators acting on the manipulator joints.

The configuration of the control scheme in Figure 4-2 is motivated by computation time requirements. Feedforward torque predictions are performed outside of the feedback loop. This allows for the learning and prediction tasks to be executed at a relatively low frequency (if necessary). The feedback term operates at higher frequency to guarantee stability [28].

The proposed scheme has to perform two main tasks, which are observing the system dynamics and controlling the manipulator. The Sections 4-2 and 4-3 discuss how these tasks are carried out in the proposed framework.

4-2 Observing

Observing the dynamics of the system is subdivided into three parts. Subsection 4-2-1 describes how the system state is derived from measurement data. The identification of linear system parameters is described in Subsection 4-2-2. Finally, Subsection 4-2-3 deals with management of training data in relation to the GPR algorithm.

4-2-1 State filter

The proposed control action, given in Equation (4-2), is dependent on angular positions, velocities and accelerations. Generally, only angular positions are directly available from the encoders of the manipulator. As a result, the velocities and accelerations are to be derived in order for the state to be fully defined.

To obtain the angular velocities from the measured positions, a first order difference equation is used, namely: $\dot{\boldsymbol{q}}(k) = s\ (\boldsymbol{q}(k) - \boldsymbol{q}(k-1))$, where k is an integer value which refers to a time step and the signal is sampled with a sampling rate s. The angular accelerations can be derived analogously as: $\ddot{\boldsymbol{q}}(k) = s\ (\dot{\boldsymbol{q}}(k) - \dot{\boldsymbol{q}}(k-1))$.

The assumption is made that the encoders are sufficiently accurate and high sampling rates can be achieved. In this case the angular velocities and accelerations can be derived relatively accurately. To deal with measurement noise, a *moving average filter* is proposed. The filter introduces a delay in the velocity and acceleration data. This is dealt with by compensating the delay of the associated torques, positions and velocities manually to obtain an equal delay in all gathered data.

4-2-2 Parameter observer

Given that the state of the device (containing $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$) is available, only the linear parameters related to the unknown object are to be observed in order to obtain the complete rigid body dynamics (RBD) description in the notation $\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\, \boldsymbol{w}(t)$. The EKF is used to this end. The implementation is equivalent to the implementation described in Subsection 3-1-2.

Note that the Kalman filter is *not* used to solve a dual problem, where the linear parameters, as well as the state are to be derived in the same framework. The separation of these tasks results in a less complex implementation and prevents the possibility of incorrectly attributing modeling errors to the measurement data.

Furthermore, the parameter identification is included for the sake of model accuracy and learning speed, not for accurately deriving true or realistic model parameters. The errors that are introduced through uncertainties in the manipulator model prevent the EKF from observing the exact parameters, however, the observed parameters give a general description of the dynamics in known and unknown parts of the state-space. The aim is to converge rapidly towards these values and subsequently remain close to these values without being influenced significantly by dynamic phenomena that are not induced by the attached object.

Some considerations are to be made with respect to the tuning of the EKF. The tuning is done taking the statements above into account. The initial process state $\boldsymbol{w}_0$ (which is the linear parameter vector) and the initial state covariance $\boldsymbol{P}_0$ are to be declared. Furthermore, the covariance matrices of the process noise $\boldsymbol{M}$ and measurement noise $\boldsymbol{N}$ need to be defined. The sigmoid function that is used to bound the linear parameters requires a lower bound, upper bound and slope ($\boldsymbol{b}_l$, $\boldsymbol{b}_u$ and $\boldsymbol{b}_s$ respectively) to be defined.

In the proposed configuration, the measurement covariance $\boldsymbol{N}$ is based on expected model errors. A higher noise covariance implies larger model errors. As a result, the identified parameters will tend towards values which generalize better and neglect unstructured local errors. The off-diagonal entries of this matrix are set to zero which reduces the effect of errors on individual joints on the parameter vector.

The initial state covariance $\boldsymbol{P}_0$ and the sigmoid function slope $\boldsymbol{b}_s$ directly influence the speed of convergence of the parameters after initializing the EKF and are tuned with respect to the chosen measurement noise covariance $\boldsymbol{N}$. A balance is to be found between speed of convergence and the extent to which a generally applicable parameter vector is found (a slower convergence prevents the parameters from fitting to local modeling errors).

The process noise covariance matrix $\boldsymbol{M}$ is added which ensures that the linear parameters do not converge toward a steady state, but are able to adapt. This is in accordance with the fact that the parameters are not perfectly valid over the entire state-space. Additionally, this also allows for *adaptation to gradual system changes.*

The lower and upper bounds for the parameter vector are set to values that can reasonably be expected to represent the unknown object including the modeling errors. Note that negative values for the lower bounds ($\boldsymbol{b}_l$) are for this reason a valid assumption (the model of the manipulator may overestimate the mass of the true manipulator). The upper bounds ($\boldsymbol{b}_u$) are chosen marginally higher than the maximum expected values, to prevent the Jacobian of the measurement update (see Equation (3-5b)) to completely go to zero.

4-2-3 Data management

The EKF is used to observe the linear parameter vector which, in combination with the regressor, contains information of the system dynamics. Dynamics that are not described by this model cannot be captured in the parameters and are therefore unobservable. These dynamics are to be captured by the data-driven Gaussian process (GP).

The GP is used to obtain a spatial description of the system dynamics. The observed dynamics are captured in the training data that is gathered over time. With reference to Equation (3-13), the applied torques are gathered in an array $\boldsymbol{\tau}$ and the measurement data obtained while applying these torques is used to configure the covariance matrix $\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} \in \mathbb{R}^{n \times n}$. The data is included in the posterior distribution through the term $\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2 \mathbb{I}\right)^{-1} \in \mathbb{R}^{n \times n}$.

During online operation, the training process is controlled by managing the incoming data. Not all measurements are used in the training data covariance matrix, since this would imply that the covariance matrix grows for every new measurement without limits. Sparsification is proposed through the use of $m < n$ *inducing variables*, which restricts the dimensionality of the GP [29]. The data is consequently stored in $\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{u}} \boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1} \boldsymbol{K}_{\boldsymbol{u},\boldsymbol{x}} \in \mathbb{R}^{m \times m}$, which limits the computational efforts when the training data set grows large. More information on this method for sparsification is found in [29]. For simplicity, the general notation for describing GPR is used in this thesis (see Section 3-2).

For long running experiments older data is sequentially removed from the torque and state measurement arrays. This prevents the computation time from increasing too much. Additionally, forgetting old data serves the purpose of dealing with *gradual time-varying system dynamics.* It is assumed that new data holds more relevance than old data. Deleting the old data therefore ensures that predictions are not based on invalid data.

4-3 Controlling

In Chapter 3, approaches are described that can be used to model the inverse dynamics of a manipulator. The parametric approach (which is based on the RBD model) and non-parametric approach (using GPR). Both approaches possess different characteristics. In an attempt to capture advantages of both methods, a semi-parametric model is proposed in [26].

The scheme in Figure 4-2 shows that the control action is comprised of a feedforward term and a feedback term. Next, a proposed implementation for these terms is given.

4-3-1 Feedforward control

The forward control term is based on knowledge of the inverse dynamics of the system. With reference to Equation (4-2), the forward term is constructed as:

$$\boldsymbol{\tau}_{\text{ff}}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) = \boldsymbol{\tau}_{\text{man}}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d) + \boldsymbol{\tau}_{\text{gp}}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d, \boldsymbol{w}(t)) \tag{4-3}$$

Ideally, this forward term cancels out the dynamics of the manipulator, which is represented as in Equation (2-2). Combining the assumed system dynamics and the control term for the *desired situation* leads to the equality:

$$\boldsymbol{M}(t, \boldsymbol{q})\, \ddot{\boldsymbol{q}} + \boldsymbol{C}(t, \boldsymbol{q}, \dot{\boldsymbol{q}})\, \dot{\boldsymbol{q}} + \boldsymbol{G}(t, \boldsymbol{q}) + \boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau}_{\text{man}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) + \boldsymbol{\tau}_{\text{gp}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{w}(t)) \tag{4-4}$$

The subscript d is dropped for the sake of legibility. The term $\boldsymbol{\tau}_{\text{man}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ represents a control torque which is based on the operator's (inaccurate) knowledge of the manipulator. The derivation of this term is based on the RBD model with estimated dynamic and kinematic parameters (captured in $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{G}}$), which is symbolically represented as:

$$\boldsymbol{\tau}_{\text{man}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \tilde{\boldsymbol{M}}(\boldsymbol{q})\, \ddot{\boldsymbol{q}} + \tilde{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\, \dot{\boldsymbol{q}} + \tilde{\boldsymbol{G}}(\boldsymbol{q}) \tag{4-5}$$

With reference to Equation (4-4), the GP consequently has to deal with the inaccuracy of the model parameters and any other unmodeled and time-varying behavior, including $\boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. The predicted torques are equal to the mean of the posterior distribution $\boldsymbol{\tau}_*$. Hence, the torques are described as:

$$\begin{aligned} \boldsymbol{\tau}_{\text{gp}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{w}(t)) &= \text{mean}\left(\mathcal{N}\left(\bar{\boldsymbol{\tau}}_*\,,\, \text{cov}(\boldsymbol{\tau}_*)\right)\right) \\ &= \boldsymbol{\Phi}(\boldsymbol{X}_*)\boldsymbol{w}(t) + \boldsymbol{K}_{*,\boldsymbol{x}}\left(\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}} + \sigma_n^2 \mathbb{I}\right)^{-1} (\boldsymbol{\tau} - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{w}(t)) \end{aligned} \tag{4-6}$$

In this definition, the variance of the noise is represented by σ_n^2 and the matrix $\boldsymbol{X}$ contains all measurement data (which is gathered as described in Section 4-2). The corresponding torques are stored in $\boldsymbol{\tau}$. For the mean of the prior distribution, the deterministic parametric function $\boldsymbol{\tau}_{\text{mean}} = \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{w}(t)$ is used, resulting in a *semi-parametric* description of system dynamics. Although [26] proposes the use of the stochastic parameter vector, the deterministic version is deemed more suitable for this application. This is motivated by the fact that a time-varying parameter vector is provided by the EKF, whereas the GP gives a spatial description. The covariance matrix of the prior distribution is created based on the *squared exponential covariance function*. More information on the subject is written in Section 3-2.

4-3-2 Feedback control

A straightforward method for control is proportional-derivative (PD) feedback control. This form of control is often used when joint-space control is considered. The same setup of using computed torque control combined with PD feedback control is applied in several researches [13, 17, 30]. For the considered type of manipulator, the feedback control term can easily be written as a summation of the error of the angular position and error of the angular velocity. Scaling the errors with the *proportional gain* $\boldsymbol{K}_p$ and *derivative gain* $\boldsymbol{K}_v$ leads to the control action:

$$\boldsymbol{u} = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} \tag{4-7}$$

Here, $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$ and $\dot{\boldsymbol{e}} = \frac{\mathrm{d}\boldsymbol{e}}{\mathrm{d}t} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$.

In this thesis, the used gains $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are diagonal. Hence, no coupling is introduced through the feedback term. The feedback for each joint is strictly independent of other joints. This prevents inaccuracies in one joint to actively disturb the control action on other joints.

Furthermore, in accordance with the philosophy of this thesis, low-gain feedback is desired (see Subsection 1-2-1). This is taken into account when setting the gains.

4-4 System change detection

The semi-parameteric control and identification approach presented in Sections 4-1 to 4-3 is able to correctly describe the inverse dynamics of a system under time-invariant conditions. However, as indicated in the problem statement (Subsection 1-2-3), the constructed scheme has to deal with time-varying phenomena. Whereas gradual changes over time are automatically compensated for by both parametric and non-parametric parts (see Subsections 4-2-2 and 4-2-3), abrupt changes in the system dynamics require a more specific procedure to retain accurate system knowledge.

4-4-1 The relevance of system change detection

In this thesis, an interest is taken specifically into manipulators that are handling various unknown objects. The picking and placing of objects by a manipulator is modeled as a discrete event. It is required that the discrete events are detected and that a switching procedure is executed to achieve rapid adaptation toward the new situation.

A detected system change has consequences for the parametric, as well as the non-parametric parts of the considered scheme. The parameter identification, which is managed through the EKF, operates on a relatively long time scale, meaning that the parameters do not converge rapidly towards their new values after a system change has occurred. The EKF is *reinitialized* after a detection to solve this problem. Detection of system changes is however more crucial for the spatial description defined through the GP. Including data from two different system descriptions into one training data set leads to incorrect predictions that are based on data from both the current and previous system. After detection, newly acquired training data should therefore be stored in a *different data set*, that is used to describe only the newly obtained system.

Dealing with such events can be done by making use of *multiple models*. Using a discrete number of different models allows for the control algorithm to make use of knowledge that is available from previous events. *Model for Sensorimotor Learning and Control* (MOSAIC) is an algorithm for multiple model control that uses pairs of forward-inverse models to describe different contexts [15]. Although the concept is promising, the used models are linear. Therefore, direct implementation of the approach is not possible. Based on the concept used in [15], an algorithm for *nonlinear* multiple model control is proposed in [16]. This research uses the *locally weighted projection regression* algorithm to describe multiple input-output relations in parallel. This method however does not make use of prior knowledge and as such, is entirely dependent on the availability of data to be able to detect system changes. This is where the semi-parametric GPR has the advantage, since it allows for system changes to be detected also when no data is directly available.

In known parts of the state-space, the GPR can be utilized to determine whether a system change has occurred. The GPR framework is ideally suited to provide a system change detection measure due to its stochastic nature. With access to the active training data set $\boldsymbol{X}_{\text{active}}$, torques can be predicted for any desired test input $\boldsymbol{x}_*$. Using a recent state measurement $\boldsymbol{x}_* = \boldsymbol{x}_i$ as test input, the *likelihood* of the corresponding torque $\boldsymbol{\tau}_i$ can be determined with respect to the posterior distribution of the active GP model. As such, the detection measure may include the probability: $p\left(\boldsymbol{\tau}_i \,|\, \boldsymbol{x}_i\,, \boldsymbol{X}_{\text{active}}\,, \boldsymbol{\tau}_{\text{active}}\right)$, which is easily determined with respect to the posterior distribution of the trained GP.

For this thesis it is deemed more interesting to focus on the issues related to system change detection in unknown parts of the state-space. This is the topic of the following subsection.

4-4-2 Model based detection in unknown parts of the state-space

In order to determine whether an abrupt change in system dynamics has occurred while in an unexplored part of the state-space, the RBD model is utilized. The forward model can be used to check whether (with the current linear parameter set) the expected state resembles the measured state while applying a certain load. Equivalently, as proposed in [16], one might use pairs of input-output data and simply use the available inverse dynamics model to verify the validity of the current parameters.

In this thesis, the RBD model is specifically used to give a general description of the dynamics of an unknown object which is engaged at the end effector of a manipulator. Unmodeled manipulator dynamics are not accurately captured by this parametric term and should therefore ideally not trigger a switching event. The degree to which a currently active parameter set is valid should for this reason be rated with reference to the ability of *any* parameter set (given the model) to fit the data at that instance. Hence, a *model based* detection measure $\eta_{\text{mb}}\left(\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{active}}, \boldsymbol{e}^{\boldsymbol{\tau}}_{\text{instant}}\right)$ is used, where $\boldsymbol{e}^{\boldsymbol{\tau}}$ is the *torque prediction error*, which is defined as the difference between the calculated torques $\tilde{\boldsymbol{\tau}}$ and the true torques $\boldsymbol{\tau}$. $\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{active}}$ is the torque prediction error obtained using predictions with the currently active parameter set. $\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{instant}}$ is obtained when using the parameter set that is computed for that specific instance.

The current torque prediction error is found using the linear representation of the dynamics, with the error described as $\boldsymbol{e}^{\boldsymbol{\tau}} = \tilde{\boldsymbol{\tau}} - \boldsymbol{\tau} = \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{w}}(t) - \boldsymbol{\tau}$. With reference to a known set of input and output data, the current model error can therefore be calculated using simply the estimate of the linear parameter vector, indicated by $\tilde{\boldsymbol{w}}(t)$.

The torque prediction error related to the instantaneously derived parameter set ($\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{instant}}$) is determined using the same approach, but now using the linear parameters $\boldsymbol{w}_{\text{instant}}$. These parameters *can* be derived using recent measurement data and the *pseudo-inverse* of the regressor $\boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. However, this method does not restrict the parameters in any way and may therefore lead to unrealistic parameters that do not represent the model as intended. It is more appropriate and convenient to use a framework that is already in place, which is the EKF. A parameter set $\boldsymbol{w}_{\text{instant}}$ is estimated by an EKF, in parallel to the currently active parameter set, but different tuning settings are used to realize fast adaptation. The utilization of this method is similar to the approach described in [31], where an adaptive secondary model is proposed which rapidly adapts to acquire the best model fit in the current situation.

With the active and instantaneous torque predictions described as above, the following system change detection measure is proposed:

$$\eta\left(\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{active}}, \boldsymbol{e}^{\boldsymbol{\tau}}_{\text{instant}}\right) = \alpha \frac{\|\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{active}}\|_1^2}{\|\boldsymbol{e}^{\boldsymbol{\tau}}_{\text{instant}}\|_1} \tag{4-8}$$

The 1-norm is preferred over the 2-norm to prevent the errors occurring at individual joints from dominating the outcome of the norm (and therefore the detection). Furthermore, the norm of the active torque prediction error appears squared in the equation to increase the influence of higher errors in the currently active parameter set. α is a tuning constant that is scaled depending on the desired sensitivity of the approach. When the detection measure reaches a certain threshold, a switch event is triggered.

4-5 Discussion

4-5-1 Application possibilities

The description of the system dynamics consists of a non-parametric part which deals with the unmodeled dynamics. The goal of the parametric description is to model the dynamics that are subjected to instantaneous changes. The instantaneous change is in this thesis represented by an object which is attached at the end effector of the manipulator. Focusing on this specific situation is however by no means a restriction. In theory, assuming the RBD model holds, the entire manipulator can be described using the linear representation $\boldsymbol{\tau}_{\text{man}} = \boldsymbol{\Phi}_{\text{man}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{w}_{\text{man}}$ [7]. When any of these dynamics are expected to experience abrupt changes, they can be appended to the linear parameter vector. Caution is required when this is attempted however, since some of the parameters may interfere with each other, which has a negative effect on the generalization capability of the parametric model.

4-5-2 Recognizing external influences

Accurate knowledge of the system is especially important for impedance controlled manipulation and for tasks that require gravity compensation. In the case of gravity compensation, interaction with an operator is to be expected. Such interactions will influence the ability of the EKF to accurately identify the linear parameters of the system, since these parameters are based on the relation between applied or measured torques and the system state. An external influence will disturb this relation, resulting in diverging parameter estimates or undesired switch events.

Several solutions could solve the described issue, for example by using camera or infrared imaging to detect the cause of an interference. Alternative solutions are methods that detect physical contact such as a sensitive skin, like on the Bosch Apas (see [5]). Or, devices that have an active switch or button to indicate interaction, like on the Baxter robot [32]. Applying one or several of these methods would suffice, however, not all robots are equipped with such measures. Measures may also be taken on the software side. The error in a torque prediction may indicate a system change, however, when the resulting error does not comply with a physical object (for example when high forces are observed in the horizontal plane), one may assume that external forces are applied.

Chapter 5

Evaluation

In accordance with the final objective formulated in the introduction, the proposed solutions are to be implemented and evaluated. To test the proposed methods on their ability to achieve high modeling accuracy and adaptivity, simulation experiments are designed and carried out. The simulations are based on a model of the Kuka youBot arm (specifications found in [33]).

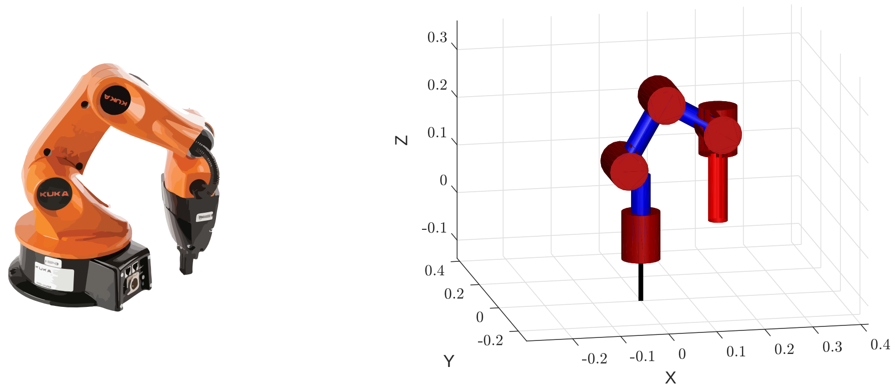
Both the simulation, as well as the lab experiments make use of the Matlab software package [34]. Within Matlab, there is made use of several toolboxes. For modeling the kinematics and dynamics of a serial manipulator, the *Robotic toolbox* [35] is used. Also the *gpml-toolbox* [36] is used for training and prediction in the Gaussian process regression (GPR) framework. The implementation of the extended Kalman filter (EKF) is arranged by the *EKF/UKF toolbox* [37].

This chapter continues on Section 5-1 with a description of the setup that is used for the simulation. The control scheme that is proposed is evaluated in Section 5-2. Subsequently, the detection and switching methods are tested in Section 5-3.

5-1 Simulation setup

The control scheme is applied on a manipulator that is designed to mimic the Kuka youBot arm, pictured in Figure 5-1a. The Kuka youBot arm is a 5-degree-of-freedom (DoF) manipulator using the articulated layout. Optionally, the manipulator is mounted on a driving base, however, the base is not considered in this thesis. A graphical representation of the Matlab simulation model is shown in Figure 5-1b. Note that the used model is not entirely representative of the youBot arm, but only based on its design. The used kinematic and dynamic parameters are loosely based on the parameters specified in in [33]. The model for this device is used for the following reasons:

- It is based on a real robot, which makes it representative in terms of joint movements and required torques.



(a) Kuka youBot 5-DoF arm[1]

(b) Simulation arm in Matlab

Figure 5-1: Kuka youBot manipulator and a model of its kinematics. The visual representation of the Matlab model is produced with the Robotic toolbox [35].

- Although it has five DoFs, three of its joints operate in the vertical plane, making the data easily interpretable.
- A Kuka youBot is available in the Delft Center for Systems and Control (DCSC) lab, which facilitates potential follow-up studies

This section continues with Subsection 5-1-1 on the description of the true system and the estimated model which are used for simulation. Additionally, definitions are given with respect to the considered rigid objects in Subsection 5-1-2. In Subsection 5-1-3, a noise model is constructed that is used to mimic true measurement data.

5-1-1 Manipulator description

For the Matlab simulations a *true system* and an *estimated model* are required. For the true system, the exact dynamic relations are assumed to be unknown for control purposes Additionally, an unknown object may or may not be attached to this device. The estimated model represents the incorrect estimate of the true system and is used by the controller to derive control torques with reference to a desired trajectory. The discrepancies in the dynamic descriptions of these two models, in addition to the influences of the unknown attached object, are to be captures by the semi-parametric control term, as stated in Equation (4-3).

In the Matlab simulations, the Robotic toolbox is used to model both true and estimated systems according to the rigid body dynamics (RBD) framework. The joint angels are defined to be all zero in the straight up position. Consequently, the model in Figure 5-1b is displayed in the pose: $[0°, 30°, 90°, 60°, 0°]$, where the entries represent the joint angles for joints 1 through 5 respectively, counted from the base towards the end effector. Furthermore, the *z-axis* of the end effector is defined to be in the axial direction of the last joint. When the final

[1] Source: https://4.imimg.com/data4/LM/VV/MY-1955060/kuka-youbot-500x500.png

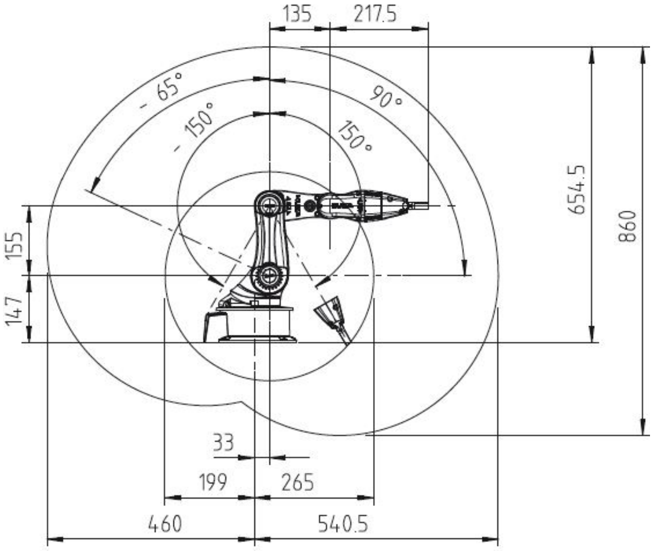


Figure 5-2: General dimensions and workspace of the Kuka youBot manipulator. Image adopted from [33].

joint is at a 0° angle, the *x-axis* points in the vertical plane and the *y-axis* is perpendicular to the vertical plane. A general indication of the dimensions and workspace of the youBot manipulator is depicted in Figure 5-2. Several adaptations to the true system are made to introduce modeling errors. This introduces errors in the forward control torque applied to compensate the true system's dynamics. The implemented discrepancies are described in the listing below.

Discrepancies in actuation

In the introduction to this report it is mentioned that innovative actuators may prove difficult to model accurately (see Subsection 1-2-2). Consequently, the torque acting on the system may not entirely represent the torque that is desired by the implemented controller. The relation $\tau_j^{\text{true}} = \tau_j^{\text{applied}} - \text{sign}(\tau_j^{\text{applied}})(\beta_j^{-1}\tau_j^{\text{applied}})^2$ is used to induce actuation related errors, where the subscript j indicates the considered joint. Using the appropriate values for β, the quadratic term causes a slightly reduced torque to be engaged at the manipulator joints.

Discrepancies in system parameters

The behavior of the system to an applied torque is described by the kinematic and dynamic system parameters. These parameters are set to different values for the estimated and true models. The used parameters are quantified in Appendix B. Additionally, a nonlinear position dependent friction term is added for the true system (i.e.: $B_j = B_j(q_j)$) to introduce local nonlinear behavior in the dynamic relations.

Discrepancies in state measurements

The state of the true system is never directly measurable. Only noisy angular position data is available. The true state is retrieved using methods described in Section 4-2. Additionally an inaccuracy in the true system is introduced by making the assumption that the system is not perfectly calibrated. Errors in the calibration of the manipulator are modeled by adding modest offset angles to the joints in the system kinematics, also visible in the Denavit-Hartenberg (DH) parameters of the model.

5-1-2 Rigid object specifications

For simulating the picking and placing tasks which are executed by the manipulator, rigid objects are defined. The objects are fully defined by the mass, center of mass and the moments of inertia as indicated by Equations (2-3a) to (2-3c). The moments of inertia are defined with reference to the center of mass and the center of mass is measured from the attachment point of the object.

Two specifications are considered while defining the parameters of the objects: the mass of the used manipulator and the payload-to-weight ratio of modern manipulators. Considering a payload-to-weight ratio of 50 % and with the mass of the simulated youBot arm being approximately 5 kg, an object is configured with a mass of 2.5 kg. Additional objects are defined for switching purposes. The parameters of the objects are displayed in Table 5-1. The center of mass is defined with reference to its attachment point.

Table 5-1: Objects used to represent instantaneous unknown system changes

	mass [kg]	Center of mass [m]	Moments of inertia [kg m^2]
Object 1	2.5	$\begin{bmatrix}0.05 & 0.05 & 0.08\end{bmatrix}$	$\text{diag}(\begin{bmatrix}1.9 & 1.9 & 1.9\end{bmatrix} \cdot 10^{-2})$
Object 2	2.0	$\begin{bmatrix}0.02 & 0.02 & 0.04\end{bmatrix}$	$\text{diag}(\begin{bmatrix}5.0 & 5.0 & 5.0\end{bmatrix} \cdot 10^{-3})$
Object 3	1.0	$\begin{bmatrix}0.01 & 0.03 & 0.07\end{bmatrix}$	$\text{diag}(\begin{bmatrix}4.2 & 4.2 & 4.2\end{bmatrix} \cdot 10^{-4})$

5-1-3 Noise model

The considered system is identified by observing the relation between the applied torques and the system state: $\boldsymbol{\tau} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. To fully identify this relation, access to the applied torques and angular positions, velocities and accelerations of the manipulator is required. Generally, the angular position is directly obtained as an output of the encoder and the velocities and accelerations must be derived from this data. In contrast to simulation data, real measurement data is subjected to noise, which can have a severe influence on the ability to derive the angular acceleration. Because of the crucial role of the measurement noise in finding the accelerations, a noise model is used in the simulation to obtain relevant results.

The noise model is based on data that is gathered from a *Kuka LBR iiwa 7 R800* at the DCSC robotics lab. This data set is assumed to be representative of currently available sensors and encoders. Two seconds of measurement data (gathered at 1000 Hz) is displayed in Figure 5-3. Differentiating and filtering the data is realized as proposed in Section 4-2. The filter is applied after both differentiations and averages over the previous 20 data samples, resulting in a 10 ms delay on the velocity and a 20 ms delay on the acceleration. The delay is compensated for when the data is prepared for parameter identification and training of the Gaussian process (GP).

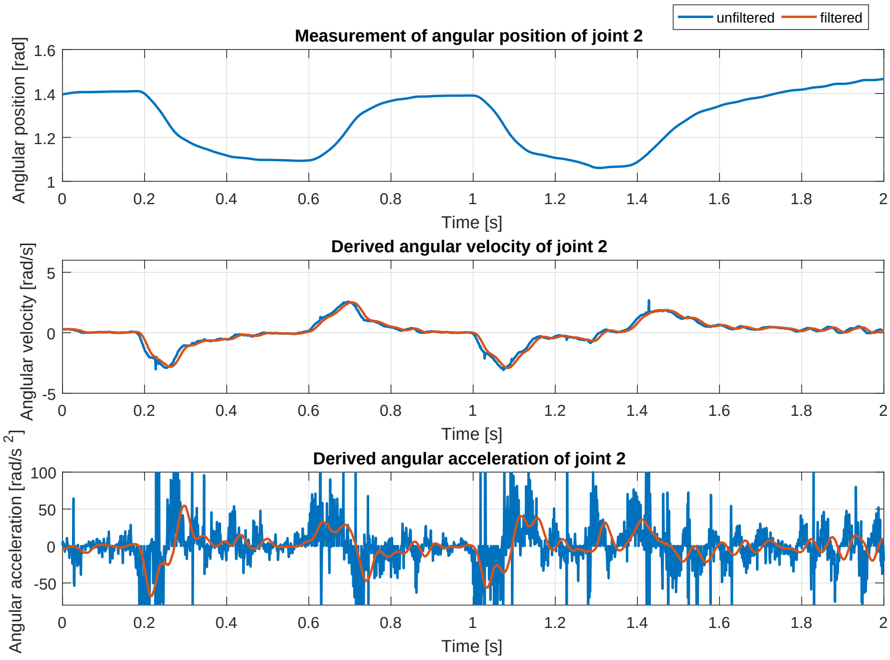


Figure 5-3: Visualization of Kuka iiwa measurement data and the derived velocity and acceleration. A moving average filter is used to retrieve the signals from the noisy measurements. The data is obtained from the Kuka LBR iiwa 7 R800 in the DCSC lab.

For the simulation, a *zero mean Gaussian noise* is added to the output of the model (the angular positions). The noise is acting on the individual joints and no cross-correlation is assumed. The variance of this noise is modeled to mimic the noise observed in the measurements displayed in Figure 5-3. Recreating the trajectory of the measured data set, a similar response is achieved for the simulation data, which is displayed in Figure 5-4. The resulting noise model is dependent on the angular accelerations and reads:

$$\epsilon_j^q(\ddot{q}_j) \sim \mathcal{N}\left(0\,,\,\left(10^{-6}\,(1+0.5\,\mathrm{abs}(\ddot{q}_j))\right)^2\right) \tag{5-1}$$

5-2 Proposed identification and control scheme

The scheme used to identify the parameters and control the simulated system is constructed with reference to the approach described in Section 4-1. The implementation of the feedback control and forward control, as well as the tuning of the EKF are dependent on the considered system. The individual components of the scheme are arranged and tested in simulation.

5-2-1 PD feedback control

The feedback requires five gains for the proportional action and five gains for the derivative action of the proportional-derivative (PD) controller. The output of the feedback control

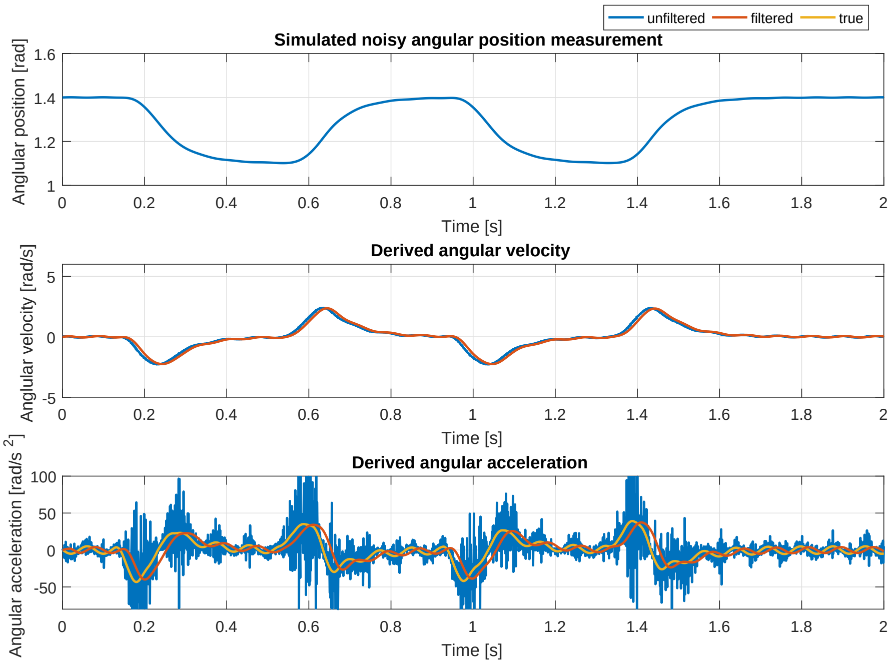


Figure 5-4: Simulation data with the added noise model. The measurement data from the Kuka device is imitated to obtain a noise model.

complements the forward control term, which provides the estimated torques to follow a desired trajectory. Consequently, using the feedback control term from Equation (4-7), the gains $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ have units $\mathrm{N\,m\,rad^{-1}}$ and $\mathrm{N\,m\,s\,rad^{-1}}$ respectively. The emphasis of this thesis is on the identification of the dynamics. The feedback merely has a supportive task. Hence, a crude method is used for the quantification of the values of $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$.

Preferably, low gains are used for the feedback in accordance with the desired compliant behavior of the manipulator. The proportional gains of joints 2, 3 and 4 are determined on the heuristic that a 15 degree deflection steady state error is accepted in the stretched most horizontal position, where the weight of the unknown object has the largest possible effect. Additionally, the maximum accepted load for the manipulator is assumed for this purpose, which is set to 2.5 kg (As stated in Section 5-1). The joints 1 and 5 are appropriately scaled with reference to the gains found for joints 2, 3, and 4. The values for $\boldsymbol{K}_v$ are simply determined as a fraction of the proportional gain and are to tuned to maintain stability. The considered values are given in Table 5-2. The same values are used during all experiments to obtain comparable results.

5-2-2 EKF parameter estimation and model based torque prediction

The equations describing the EKF (see Equations (3-3a) to (3-3f)) are implemented to derive the linear parameters for the model based description of an unknown object which is attached to the end effector of the manipulator. To determine the extent to which the EKF is able to achieve this goal in the presence of significant manipulator model inaccuracies, simulations are run.

Table 5-2: PD-Gains used for feedback controller

	K_p	K_v
axis 1	45	3
axis 2	45	3
axis 3	30	2
axis 4	18	1.2
axis 5	9	0.6

The EKF is evaluated on two separate tasks. First, the parameter estimation is tested. Subsequently, the ability of the parametric model to accurately describe the error dynamics is assessed. To gather data for the identification, both simulation are run while following a reference trajectory ($[\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)]$). The proposed identification and control scheme (without the non-parametric term) is implemented to this end. Feedback is used to manage the deviations from the desired reference trajectory, where the gains are defined as in Table 5-2. *Object 1* (see Subsection 5-1-2) is attached to the end effector of the true system from $t = 0$ onwards. The EKF is configured in accordance with the guideline proposed in Subsection 4-2-2. The tuning is at aimed at achieving rapid convergence during the initial phase of the simulation, followed by moderate adaptation to obtain a general representation of the linear parameters. The initial estimate of the linear parameter vector is set zero: $\boldsymbol{w}_0 = \boldsymbol{0}_{10\times 1}$, which implies that the controller does not expect an object to be attached to the manipulator. Furthermore, no time-varying phenomena are introduced in this simulation.

The first simulation is aimed at evaluating the ability of the EKF to identify the relevant parameters. This is done for two separate cases. First, a true system is used that corresponds exactly with the estimated model, hence, only the effects of the unknown object are to be accounted for. Subsequently, the true model is considered where inaccuracies and nonlinearities are added.

Estimates of the first four parameter (m, mc_x, mc_y and mc_z) are displayed over time in Figure 5-5. It becomes clear that rapid convergence is achieved for all considered parameters. The estimated related to m and mc_x are immediately observable and start converging from the start of the simulation. mc_y and mc_z are observed from $t \approx 2\,\text{s}$, and then converge quickly. For the *ideal case*, where the true system equals the estimated model, the estimates converge to their true values. This does not hold true for the *realistic case*. The discrepancies between the system and the model are projected onto the linear parameters, which therefore are not entirely representative of the attached object.

The estimates related to the diagonal entries of the moment of inertia matrix (I_{xx}, I_{yy} and I_{zz}) are displayed in Figure 5-6. This figure clearly demonstrates that not all parameters are observed during an individual operations. When certain parameters (in this case the estimates of I_{xx} and I_{yy}) are not excited, the estimates remain at their initial values. The parameter I_{zz} is observable in this simulation and converges to approximately the true value. The diagonal entries of the state covariance give an indication of the extend to which a parameter is observed. When a parameter is not excited, its entry in the covariance matrix remains unaffected. This relation between the parameter estimates and their respective covariance matrix entries is clearly visible in Figure 5-6.

The second simulation is aimed at evaluating the ability of the parametric model (in combina-

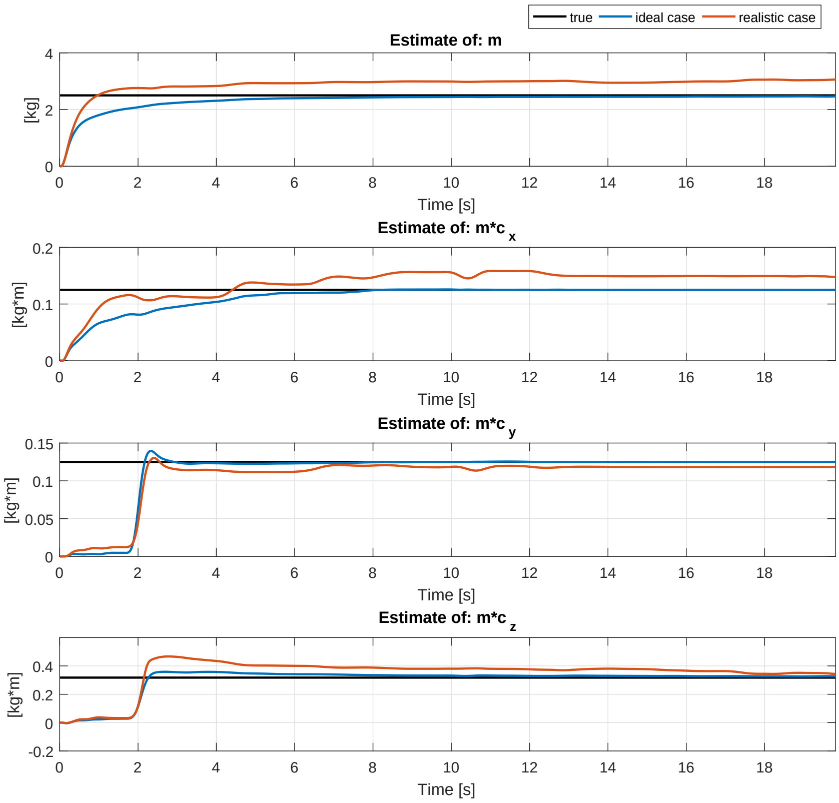


Figure 5-5: Mass and moments estimates obtained using the EKF. The identification is compared for the ideal case and realistic case, where the realistic case takes modeling errors into account. The ideal case assumes perfect knowledge. The estimates for the realistic case do not converge to the true object parameters due to the influence of the modeling errors.

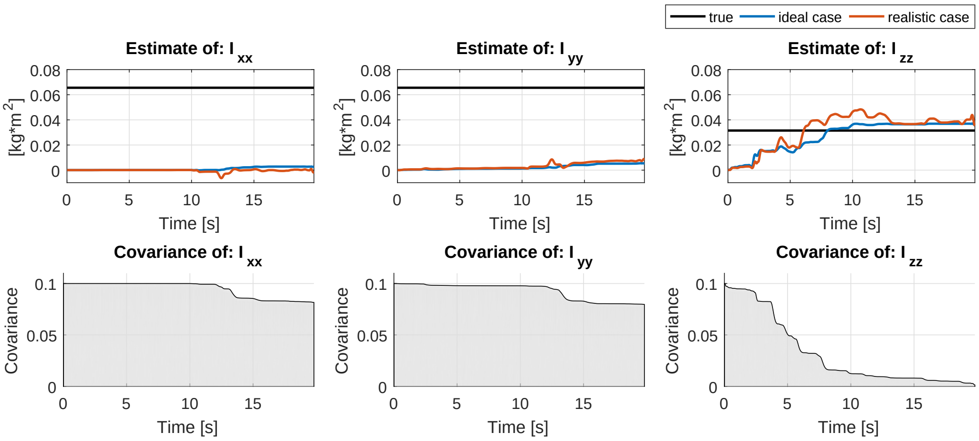


Figure 5-6: Moments of inertia estimates obtained using the EKF. The parameters I_{xx} and I_{yy} prove to be only marginally observable and do not fully converge towards their true values. I_{zz} does converge for both the ideal and realistic case, which also shows in the corresponding entry of the covariance matrix.

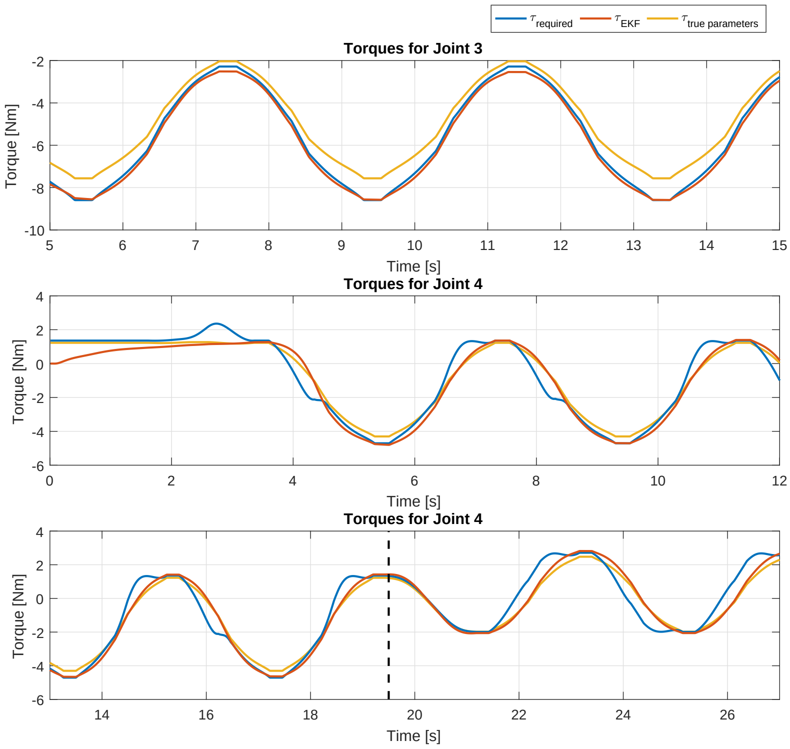


Figure 5-7: Comparison of torques while using the model based forward control scheme. The blue line indicates the torques that are required for perfect forward control. The red line represents torques obtained using the parameters identified by the EKF. The yellow line is obtained when the true parameters are used. The black dashed line indicates the start of the pose change of the second and third joints.

tion with EKF parameter estimation) to describe the desired system dynamics. A trajectory tracking task is executed that is primarily focused on the fourth joint. The angels of the first and last joint are for this task set to zero. The trajectories for the remaining joints are depicted in Figure 5-8

Figure 5-7 shows three plots that give insight into the advantages and disadvantages of this configuration. The *first plot* shows the torques for the third joint of the youBot arm. A comparison is made between the *required* torques, the torques determined with the *EKF estimates* and torques determined using the *true parameters*. It is clear that the EKF identified parameters do not accurately describe the true object (see Figure 5-5), but do achieve a better fit than when using the true parameters. This confirms the statement that the EKF can partly compensate for modeling errors. The *second plot* in Figure 5-7 compares the required and estimated torques at the fourth joint from $t = 0\,\mathrm{s}$. The EKF estimate initializes at $0\,\mathrm{N\,m}$, but quickly converges toward the required values. The remaining errors are accredited to error dynamics that cannot be captured by the parametric model. Finally, the *third plot* demonstrates the generalization capabilities of the parametric approach. A trajectory

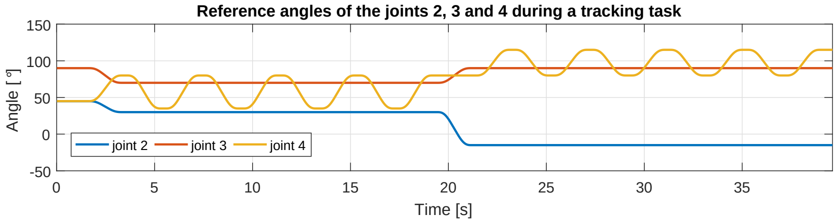


Figure 5-8: Reference angels defined for a tracking task. The task is focused on the fourth joint of the manipulator. The second and third joint transition into a new pose halfway through the simulation. The first and fifth joint remain unused.

transition occurs at $t = 19.5\,\mathrm{s}$ (indicated with the black dashed line). The parametric model does not suffer large losses in accuracy. The remaining errors are again accredited to dynamics that are not captured by the parametric model.

5-2-3 Torque prediction using GPR

Implementing the GPR framework results in a non-parametric description of the error dynamics that are induced by the discrepancies between the true system and estimated model. A zero-mean GP is used to investigate the capabilities of GPR to accurately describe these dynamics. Based on the simulation results, decisions are made regarding the specific implementation of the GPR framework.

The setup of the simulation is similar to the setup used in Subsection 5-2-2. from $t = 0\,\mathrm{s}$ forward, *object 1* is attached to the end effector of the manipulator. The zero-mean GP is used to model both the error dynamics, as well as the dynamics related to the unknown object. Low-gain feedback is used in accordance the desired compliant behavior of the system. The feedback serves the purpose of guiding the manipulator towards its desired trajectory.

Training the GP is carried out in an online fashion, as is required for the stated research goal of this thesis. The implementation of online real-time GPR is considered a complex problem which is treated in many researches (e.g. in [13]). For the simulation it is not required that the learning is to be carried out in real-time. Consequently, the computation time of the used training approach is for this simulation not restricted by the chosen sampling time of the control algorithm.

GPR setup description

The covariance matrix of the GP prior is constructed using the squared exponential covariance function with a diagonal weighting matrix. The entries on the diagonal can be interpreted as length-scales. Methods (such as *marginal likelihood maximization* or *cross-validation*) are available for optimizing these hyper-parameters. However, the optimization of these parameters on the limited amount of available data gathered during operation has not led the desired performance. Erratic behavior is observed in untrained parts of the state-space. Consequently, tuning is done manually, where the length-scales are set to values best suitable for the dynamics at hand.

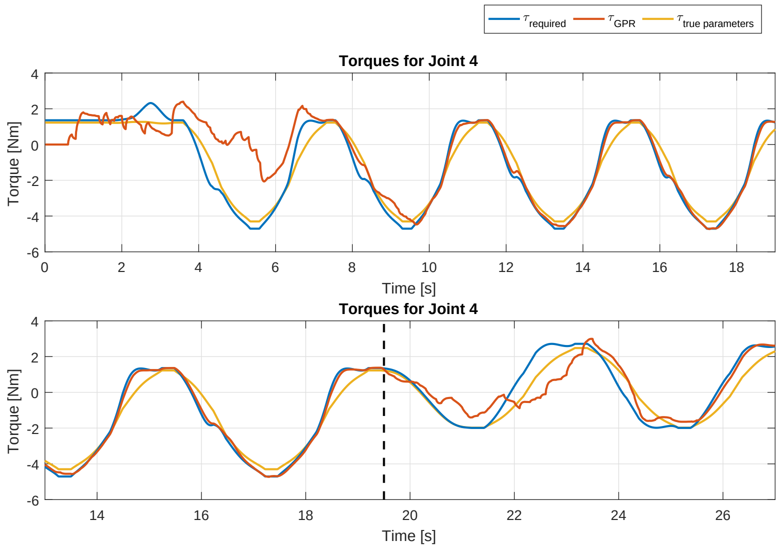


Figure 5-9: Comparison of torques while using GPR predictions. The blue line indicates the torques that are required for perfect forward control. The red line represents torques obtained by the GPR predictions. The yellow line is obtained when the parametric model is used in combination with the true object parameters. The black dashed line indicates the start of the pose change of the second and third joints.

Sparsification is implemented using *inducing variables* as indicated in Subsection 4-2-3. The locations of the inducing inputs are often determined with reference to the used sparsification method. An overview of approaches is given in [29]. Given the occurring issues with the optimization of hyper-parameters, the flexibility for the location of the inducing inputs is also reduced. The length-scales considered in the squared exponential kernel limit the placement of the inducing inputs. Taking this into account, the inducing inputs are for this thesis defined to be a *subset of the acquired measurement data*. The subset is defined by using an equally spaced distribution of the corresponding unique reference trajectory data points. The variational free energy (VFE) approach is used for approximating the posterior in the sparse GP framework. Based on [38], this approximation method shows better covariance estimations, which is deemed relevant for in this research (it reduces the possibility of an illegitimate system change detection, which is related to Section 4-4). The implementation of GPR with VFE approximation in Matlab is achieved using the GPML toolbox [36].

Evaluating simulation results

Utilizing the GPR framework, torques are inferred from data that is gathered during operation. Figure 5-9 shows the required torques compared to the torque predictions at the fourth joint of the simulation manipulator. Predictions of the parametric model with true linear parameter vector are also depicted. The *first plot* shows the predictions from $t = 0\,\text{s}$ to $t = 19\,\text{s}$. The predictions based on the GP are clearly not accurate during the initial phase of the tracking task. However, when more data is gathered, the predictions become more similar to the required torque.

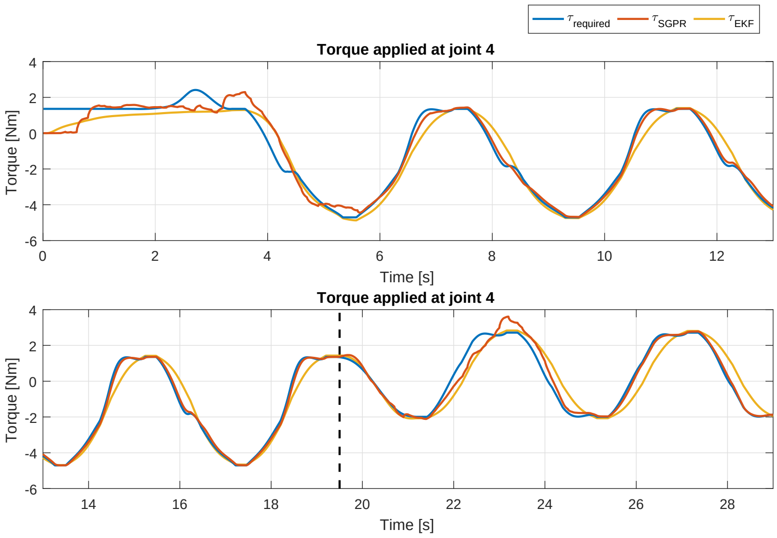


Figure 5-10: Comparison of torques while using GPR predictions in combination with EKF parameter identification. The blue line indicates the torques that are required for perfect forward control. The red line represents torques obtained using the semi-parametric model (here referred to as SGPR) for predictions. The yellow line is obtained when only only the parametric model is used in combination with the EKF identified parameters. The black dashed line indicates the start of the pose change of the second and third joints.

The *second plot* of Figure 5-9 shows the application of the GP attempting to predict torques during and after a new part of the state-space is entered. The GP model does not generalize to unknown parts of the state-space, which drastically impacts the ability of the GP to predict the required torques at these locations. As a result, predictions tend to drift towards the mean of the prior. As new data is gathered the predictions become more accurate again. Clearly, the availability of data in the considered part of the state-space is vital to the accuracy of the torque predictions.

5-2-4 Combined GPR with EKF parameter estimation

From the simulation results presented above, it is concluded that the parametric RBD model in combination with the parameter estimation from the EKF describes the true system accurately as long as the error dynamics can be captured by the parametric model. The method shows good generalization capabilities. A non-parametric approach is well suited to deal with unmodeled dynamics. In the following simulation, the *semi-parametric* scheme is used. The primary motivation for combining the parametric and non-parametric methods is to benefit from the opportunities presented by both individual methods. The settings used for the EKF and GPR implementations are identical to the settings used in the previous experiments.

In the *first plot* of Figure 5-10, the simulation is again initialized with no knowledge of the linear parameters. Furthermore, no data is available for the GP at $t = 0\,\mathrm{s}$. Comparing Figure 5-10 to Figure 5-9, it is clear that the the semi-parametric method provides a more accurate description compared to the GPR. The linear parameters that are identified by the

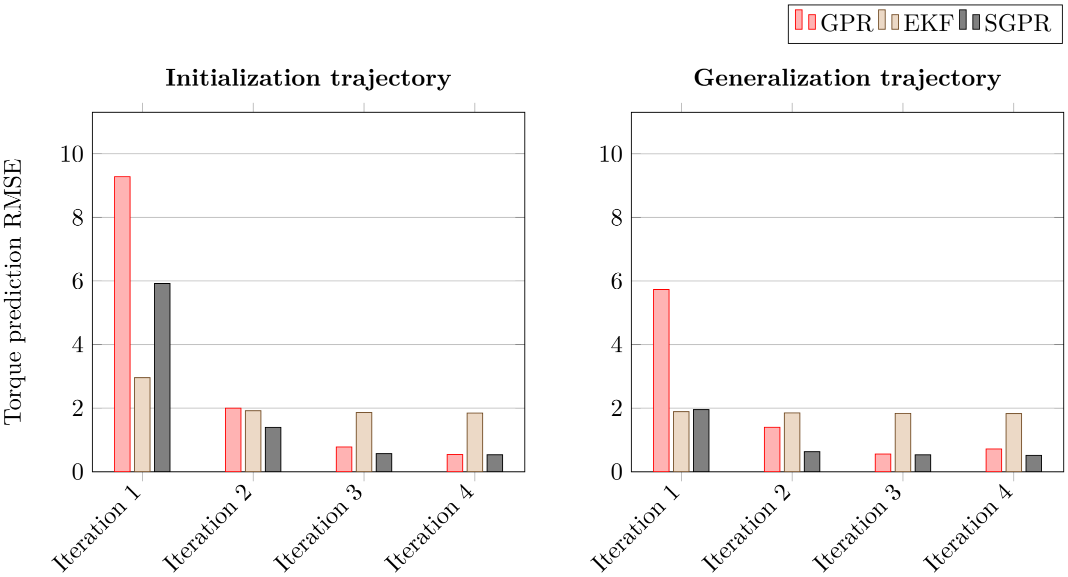


Figure 5-11: Comparison of modeling methods based on the torque prediction RMSE. Results are obtained on initialization and generalization properties.

EKF give shape to the mean function of the GP, which ensures fast convergence of the torque predictions towards the required torques. The nonlinearities that cannot be modeled by this parametric part are initially compensated for by the feedback term, after which the GP is trained on the measured data and is able to describe these nonlinearities independently. The *second plot* in Figure 5-10 shows the capability of the used scheme to deal with a transition in the followed trajectory. Comparing Figure 5-10 to Figure 5-9 again, it is evident that the semi-parametric model gives far better generalization capabilities.

The included figures are useful to configure the GP and EKF for identification, training and prediction. However, in order to quantitatively test the effectiveness of the combined approach, an additional experiment is proposed. The goal of this experiment is to compare the combined approach to the individual approaches on the three criteria specified in Chapter 4: *learning speed*, *modeling accuracy* and *generalization capabilities*.

Because all approaches require online learning to reach their full potential, the control scheme from Figure 4-2 is used here. The most relevant performance measure for the considered research goal is the difference between the required torques and the predicted torques. Alternatively, the difference between the desired and actual angular position of the controlled system can be used as a measure to quantify the tracking performance.

The experiment is constructed to allow for the three criteria to clearly be observed from the results. The experiment is therefore divided into an *initialization* part and a *generalization* part. For the initialization, a single trajectory is tracked four times successively and the torque prediction and joint position errors are determined. Subsequently, using the knowledge gained during this first trajectory, a generalization run is executed on a different trajectory. Comparing the data from the two parts shows the degree to which the considered methods meet the set criteria. To ensure independence of the order in which the different trajectories are used, each trajectory is used for both initialization and generalization. Applying this approach for three trajectories, results from three initialization and six generalization

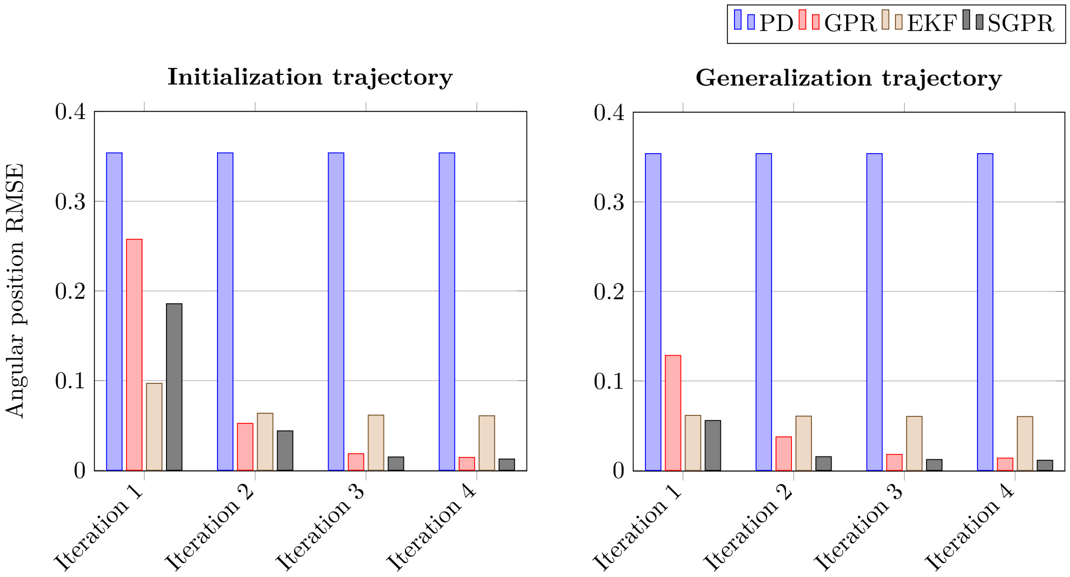


Figure 5-12: Comparison of modeling methods based on the angular position RMSE. Results are obtained on initialization and generalization properties.

experiments are gathered (each trajectory can be used twice for generalization). The root-mean-square error (RMSE) of the angular position and prediction torques are subsequently averaged to obtain the final results.

Focusing on the experiment results displayed in Figure 5-11, it is clear that the parametric model with EKF parameter estimation gives the best results for initialization. GPR cannot give accurate predictions during the first iteration. The semi-parametric GPR (or SGPR for short) remarkably does not initialize as well as expected considering that, in the absence of data, the EKF determined mean is used. This result is likely explained by the early inaccurate data. Apparently, the SGPR predictions are based on this disruptive data where it would have been more beneficial if the EKF mean was followed. The iterations two to four show a strong improvement for both GP methods. The EKF only improves slightly after the first iteration.

Continuing on the results regarding the generalization, is seen that the GPR description has improved somewhat compared to the first iteration during the initialization phase. Some of the previously gathered data has apparently been reused for this trajectory. The EKF Does not suffer from changing the trajectory and remains approximately equally accurate. The SGPR prediction behaves more predictable in this case, where it now scores approximately equal to the EKF prediction. Both GP approaches prove to give accurate predictions on the later iterations, but the SGPR method has the upper hand due to its faster convergence.

It can be concluded that the semi-parametric GPR method scores high on accuracy and learning speed and its generalization capabilities are mainly determined by the ability of the parametric mean to accurately describe relevant the dynamics.

Finally, the tracking performance is shortly discussed. Figure 5-12 shows the RMSE of the angular positions. These results are obtained from the same data as is used for Figure 5-11. Taking a closer look, the individual results, a very similar pattern can be observed in the data, as might be expected. However, in this experiment, the three forward control approaches can be compared with a feedback controlled setup. Using the forward control term clearly pays off when using low-gain feedback.

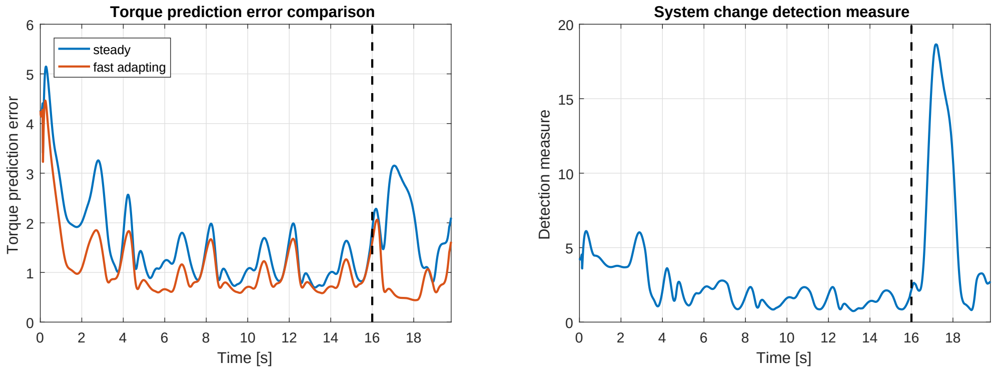
5-3 Object detection using the extended Kalman filter

The results in Section 5-2 show the ability of the proposed scheme to effectively meet the performance criteria stated in Chapter 4. The unknown object is included in the system dynamics directly during the simulation initialization. The detection of system changes was not considered in these simulations. The model based detection scheme discussed in Subsection 4-4-2 is evaluated in this section.

The considered switching scheme is able to distinguish between error dynamics that are induced by the instantaneous system changes and error dynamics that are induced by errors in the manipulator model. This is achieved by the implementation of a secondary EKF that uses a stationary state covariance matrix that is fixed to equal the initial state covariance $\boldsymbol{P}_0$. Hence, during the initialization phase, the estimates provided by this EKF show a similar response compared to the EKF used for the parametric forward control term. Whereas the primary EKF converges to steady parameters with a slow adaptation rate, the second EKF retains its rapid adaptation capabilities.

Based on the torque prediction errors obtained through both EKFs, the detection measure is computed. A moving average filter is used to decrease the influence of sporadic peaks in the error signals. The torque prediction errors corresponding to both EKFs are displayed in Figure 5-13a. In this figure it is visible that the errors follow a similar trajectory that is induced by the errors in the manipulator model. When the system change occurs (by changing from *object 1* to *object 2*, see Table 5-1), the error increases significantly when using the converged parameter set, whereas the fast adapting parameter set causes a smaller prediction error. The detection measure is shown in Figure 5-13b. Even though a delay is introduced by filtering the signals and the temporal behavior of the EKF, a clear indication for the system change is obtained.

By tuning the α-parameter and setting an appropriate threshold, the detection measure is used to force a switching event. A simulation is run where sequentially the three objects are attached to the manipulator (first *object 3*, then *object 1* and then *object 2*). Figure 5-14 shows the first four parameters from the linear parameter vector. It can be observed that the parameters converge toward a steady configuration (identical to the behavior shown in Figure 5-5). When the system change has occurred, the detection triggers the switching event. In this simulation, the state covariance matrix of the EKF is reset to stimulate fast convergence towards the new parameters. Alternatively, one might also adopt the values from the fast adapting EKF, but this results in abrupt changes in the torque predictions used for control. The detection after the second change shows a slightly longer delay as compared to the first change. This can be explained by the fact that the difference in mass is relatively small. The detection may in this case be less evident than for objects that are less similar. Note that for even more similar objects, the detection may fail and overlook the system change. If so, the prediction errors are marginal and the change may effectively be dismissed as a gradual system change.



(a) Torque prediction error for steady and fast adapting parameter sets

(b) Detection measure constructed from the torque prediction errors

Figure 5-13: Torque predictions errors and detection measure during a system change occurrence. When the system change occurs (black dashed line), the prediction errors show a diverging response. This effect is captured in the detection measure.

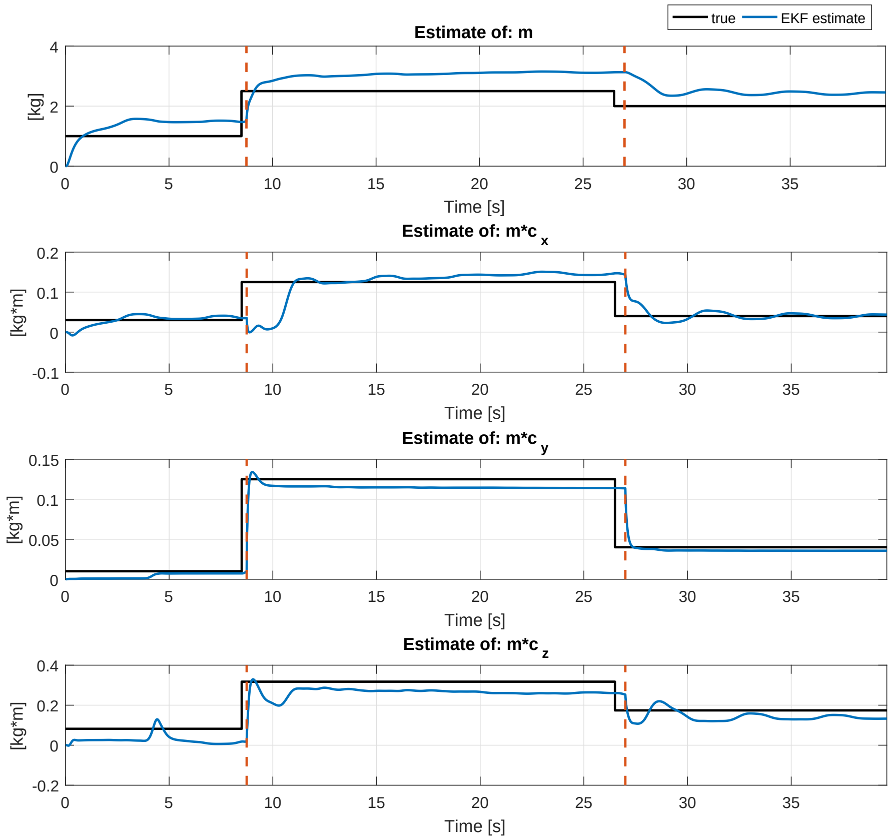


Figure 5-14: Four entries from the linear parameter vector during an operation with occurring system changes. The true system changes are observed by looking at the black lines. The red dashed lines indicate when a switching event is triggered.

Chapter 6

Conclusions and recommendations

This thesis provides an approach for identifying the dynamics of a robot manipulator which is subjected to both model uncertainties as well as (instantaneous) system changes. This is achieved by using an extended Kalman filter (EKF) to identify parameters of the rigid body dynamics (RBD) model, in combination with the data driven non-parametric Gaussian process regression (GPR) framework. The approach describes the inverse dynamics of the considered system. Instantaneous system changes can be detected using EKF parameter estimates. Conclusions can be drawn with respect to the validity, practicality, and relevance of the proposed approach. These conclusion are found in Section 6-1. In Section 6-2, recommendations are given for future research.

6-1 Conclusions

Various industries have shown an increasing interest in robot manipulators. This interest also exists for manipulators that are designed as lightweight collaborative devices. These innovative types of manipulators are programmed to cooperate with their operator and may include compliant joints or actuators to increase safety. Having accurate knowledge of the system's dynamics has beneficial effects regarding control purposes, but obtaining an accurate description is made difficult by these innovative joints and actuators. High payload-to-weight ratios further increase the influence of exogenous alterations on the device.

The research goal of this thesis is aimed at constructing a method that can deal with uncertain system dynamics and instantaneous time-varying phenomena. To this end, a semi-parametric approach is suggested in Chapter 4, where a parametric description is used to adapt rapidly towards expected changes in the system dynamics. In addition, the non-parametric description is used to deal with unmodeled dynamics, which cannot be compensated for by the parametric part. The approach is implemented in an online fashion to deal with these (instantaneous) time-varying phenomena. Simulations are run to test the capabilities of the constructed approach. Results of these simulations are included and discussed in Chapter 5. Based on the results and the process to obtain these results, several conclusions are drawn.

Using a semi-parametric approach to derive the dynamic relations of an uncertain system in a time-varying environment has a positive effect on modeling accuracy, learning speed and generalization capabilities

The three criteria that are used to test the capabilities of the semi-parametric approach are the modeling accuracy, learning speed and the extent to which the dynamics description generalizes towards different parts of the state-space. Simulation results in Section 5-2 show that, of the three considered approached, the semi-parametric approach is the only approach that achieves good performance on all three criteria.

Using a model based description of an object at the end effector has beneficial effects on the accuracy of the obtained dynamic relations

The uncertainties that are present in the manipulator model may partly be compensated for by the parameters that are used to identify the object attached to the end effector. These parameters are only locally valid, but are updated over time, resulting in a parametric description which becomes more accurate as the system remains longer in a certain part of the state-space.

Instantaneous system changes can be detected using a pair of EKFs that use fast and slow adaptation rates for the identification of a linear parameter vector

The torque prediction errors obtained when using the two separate parameter vector estimates show a diverging response when a system change has occurred. This effect is captured in a detection measure which is used to trigger a switch event.

The practicality of an online trained GPR for the application of real-time robot control is questionable

Although a considerable amount of research is performed in this field, the application of online GPR is considered to be the least robust element of the proposed learning scheme. The GPR framework requires a lot of decision making, such as defining the covariance function, hyper-parameters, and possibly optimization and sparsification schemes. These decisions have a significant influence on the performance of the learning scheme and the performance may vary between tasks. Obtaining a generally well performing setup requires a considerable amount of effort, but yields no guarantees.

6-2 Recommendations for future research

With reference to the simulation results and the conclusions that are drawn, a number of recommendations are formulated and are listed below.

Decrease the amount of heuristics used in the described approaches

Both the EKF as well as the GPR framework currently require decision making which is in this report mainly dealt with through tuning of the relevant parameters. The same holds for the method used for system change detection. Tuning these parameters is a time-consuming activity and presumably yields non-ideal and non-repeatable performance. Future research may lead to a more strict and ideal implementation of the used methods.

Investigate the possibility for hyper-parameter optimization in this framework

The hyper-parameters of the GPR are manually set in this thesis (which contributes to the heuristics mentioned in the previous recommendation). Hyper-parameter optimization can be used to find hyper-parameters that increase the accuracy of the torque predictions. However, no desirable results where obtained when predicting on data that was not used for the optimization. Further research on this topic in combination with the methods proposed in this thesis is recommended.

Extend the proposed detection method

In the current method, the system change detection is focused on detecting in parts of the state-space where no data is available. In parts where data is available it is preferred to infer system changes based on the GPR framework. Investigating the possibility to combine both approaches for detection is suggested. This may lead to a more general framework and possibly better detection capabilities in the transition area between known and unknown parts of the state-space.

Apply the proposed methods on a real manipulator

The modeling errors used in simulation may differ from the errors observed on a real manipulator. Results obtained from experiments on a real setup may indicate further areas for improvement. Preferably, a manipulator is used that fits the described characteristics mentioned in the introduction and that allows for low-level control.

Investigate how force sensor data can be utilized in this framework

In the setup considered for this thesis, the gathered data consists of applied torques and angular positions, velocities and accelerations. The angular positions are obtained through sensor measurements. Some robot devices are equipped with load sensors. Such sensor data may influence how best to implement the approaches proposed in this thesis.

Appendix A

Mathematical definitions

Alternative notation for linear operations

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad , \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad , \quad \boldsymbol{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix}$$

While working with cross products, the following transformation matrix is proposed:

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\boldsymbol{a} \times \boldsymbol{b} = S_\times(\boldsymbol{a})\, \boldsymbol{b} = -\boldsymbol{a}\, S_\times(\boldsymbol{b}) \tag{A-1}$$

Additionally, to transfer the six unique entries of a symmetric 3×3 matrix into a 6×1 vector, a transformation as in Equation (A-2) is applied.

$$\boldsymbol{La} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 \\ 0 & a_1 & 0 & a_2 & a_3 & 0 \\ 0 & 0 & a_1 & 0 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \\ L_{22} \\ L_{23} \\ L_{33} \end{bmatrix}$$

$$\boldsymbol{La} = S_I(\boldsymbol{a})\boldsymbol{L}' \tag{A-2}$$

Sigmoid function definitions

Boundary conditions on a parameter vector are applied by transforming the state through a sigmoid function.

The true state is function of the transformed state with the upper bound $\boldsymbol{b}^+$ and lower bound $\boldsymbol{b}^-$. The slope of the function is defined through $\boldsymbol{c}$.

$$\boldsymbol{x} = \mathrm{Sig}(\boldsymbol{x}') = \frac{\boldsymbol{b}^+ - \boldsymbol{b}^-}{1 + e^{-\boldsymbol{c}\boldsymbol{x}'}} \tag{A-3}$$

The transformed state is found through the inverse sigmoid function with the same bounds.

$$\boldsymbol{x}' = \mathrm{Sig}^{-1}(\boldsymbol{x}) = \frac{\ln(\boldsymbol{b}^+ - \boldsymbol{x}) - \ln(\boldsymbol{x} - \boldsymbol{b}^-)}{-\boldsymbol{c}} \tag{A-4}$$

The derivative of the sigmoid function with respect to the transformed state is written as:

$$\frac{\partial\,\mathrm{Sig}(\boldsymbol{x}')}{\partial \boldsymbol{x}'} = \frac{\boldsymbol{c}(\boldsymbol{b}^+ - \boldsymbol{b}^-)e^{\boldsymbol{c}\boldsymbol{x}'}}{(e^{\boldsymbol{c}\boldsymbol{x}'} + \boldsymbol{1})^2} \tag{A-5}$$

Appendix B

Description of simulation manipulator

The manipulator used in the Matlab simulations is based on the Kuka youBot. The kinematic and dynamic parameters are loosely based on the parameters specified in [33]. The values used in this appendix are all directly related to the values used for the implementation in the Robotic toolbox in Matlab [35].

The tables displayed below give the values used for the rigid body dynamics (RBD) description in this report. The values that have been adapted to represent modeling errors are indicated in blue. All values are displayed with *si-units*.

Table B-1: DH-parameters as used for the estimated model

	α	a	θ	d
axis 1	$-90°$	-0.034	0	0.147
axis 2	0	0.155	$-90°$	0
axis 3	0	0.135	0	0
axis 4	$90°$	0	$90°$	0
axis 5	0	0	$0°$	0.171

Table B-2: DH-parameters as used for the true system

	α	a	θ	d
axis 1	$-90°$	-0.034	0	0.147
axis 2	0	0.155	$-95°$	0
axis 3	0	0.135	0	0
axis 4	$90°$	0	$85°$	0
axis 5	0	0	$5°$	0.171

Table B-3: Motor, gear and joint parameters as used for the estimated model

	J_m	R	B	T_C
axis 1	$13.91 \cdot 10^{-6}$	156	$1 \cdot 10^{-4}$	0
axis 2	$13.91 \cdot 10^{-6}$	156	$1 \cdot 10^{-4}$	0
axis 3	$13.57 \cdot 10^{-6}$	100	$1 \cdot 10^{-4}$	0
axis 4	$9.32 \cdot 10^{-6}$	71	$1 \cdot 10^{-4}$	0
axis 5	$3.57 \cdot 10^{-6}$	71	$1 \cdot 10^{-4}$	0

Table B-4: Motor, gear and joint parameters as used for the true system

	J_m	R	B	T_C
axis 1	$13.91 \cdot 10^{-6}$	156	$1 \cdot 10^{-4} \times 1.2$	0
axis 2	$13.91 \cdot 10^{-6}$	156	$1 \cdot 10^{-4} \times 1.2$	0
axis 3	$13.57 \cdot 10^{-6}$	100	$1 \cdot 10^{-4} \times 1.2$	0
axis 4	$9.32 \cdot 10^{-6}$	71	$1 \cdot 10^{-4} \times 1.2$	0
axis 5	$3.57 \cdot 10^{-6}$	71	$1 \cdot 10^{-4} \times 1.2$	0

Table B-5: Link parameters as defined for the estimated model

	Link mass [kg]	Center of mass [m]	Link inertia [kg m^2]
axis 1	0.139	$\begin{bmatrix}0.015 & 0.500 & -0.015\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.006 & 0.003 & -0.006\end{bmatrix})$
axis 2	1.318	$\begin{bmatrix}-0.041 & 0.000 & 0.020\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0006 & 0.003 & 0.003\end{bmatrix})$
axis 3	0.821	$\begin{bmatrix}-0.030 & 0.000 & -0.010\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0006 & 0.002 & 0.002\end{bmatrix})$
axis 4	0.769	$\begin{bmatrix}0.000 & 0.015 & 0.054\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0001 & 0.0001 & 0.0001\end{bmatrix})$
axis 5	0.091	$\begin{bmatrix}0.000 & 0.000 & -0.113\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0002 & 0.0002 & 0.0001\end{bmatrix})$

Table B-6: Link parameters as defined for the true system

	Link mass [kg]	Center of mass [m]	Link inertia [kg m^2]
axis 1	0.139×1.1	$\begin{bmatrix}0.015 & 0.500 & -0.015\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.006 & 0.003 & -0.006\end{bmatrix}) \times 1.2$
axis 2	1.318×1.1	$\begin{bmatrix}-0.041 & 0.000 & 0.020\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0006 & 0.003 & 0.003\end{bmatrix}) \times 1.2$
axis 3	0.821×1.1	$\begin{bmatrix}-0.030 & 0.000 & -0.010\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0006 & 0.002 & 0.002\end{bmatrix}) \times 1.2$
axis 4	0.769×1.1	$\begin{bmatrix}0.000 & 0.015 & 0.054\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0001 & 0.0001 & 0.0001\end{bmatrix}) \times 1.2$
axis 5	0.091	$\begin{bmatrix}0.000 & 0.000 & -0.113\end{bmatrix}$	$\text{diag}(\begin{bmatrix}0.0002 & 0.0002 & 0.0001\end{bmatrix})$

Appendix C

Simulation figures

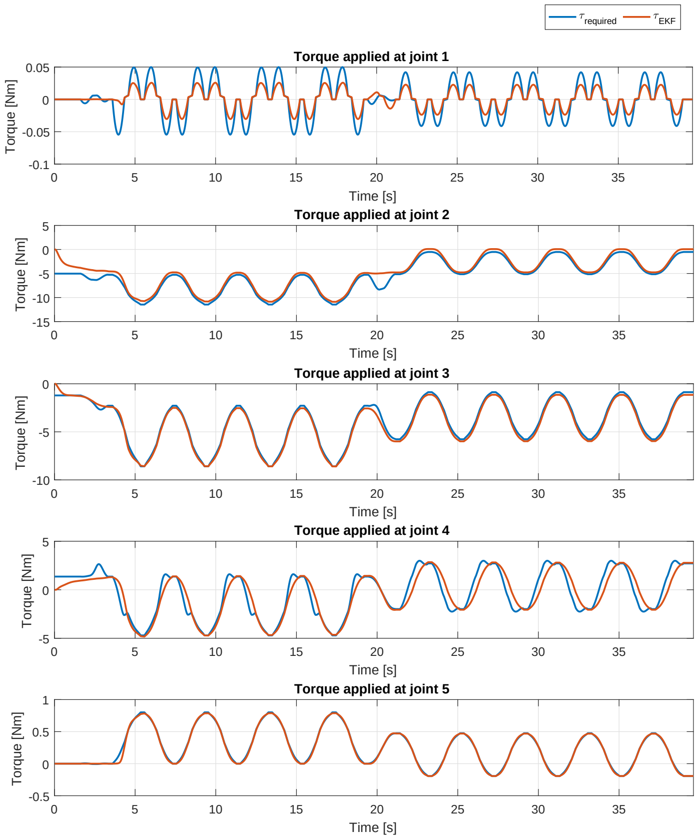


Figure C-1: Plots related to Subsection 5-2-2: Torques required for perfect tracking compared to the torques applied through the control action with EKF parameter identification

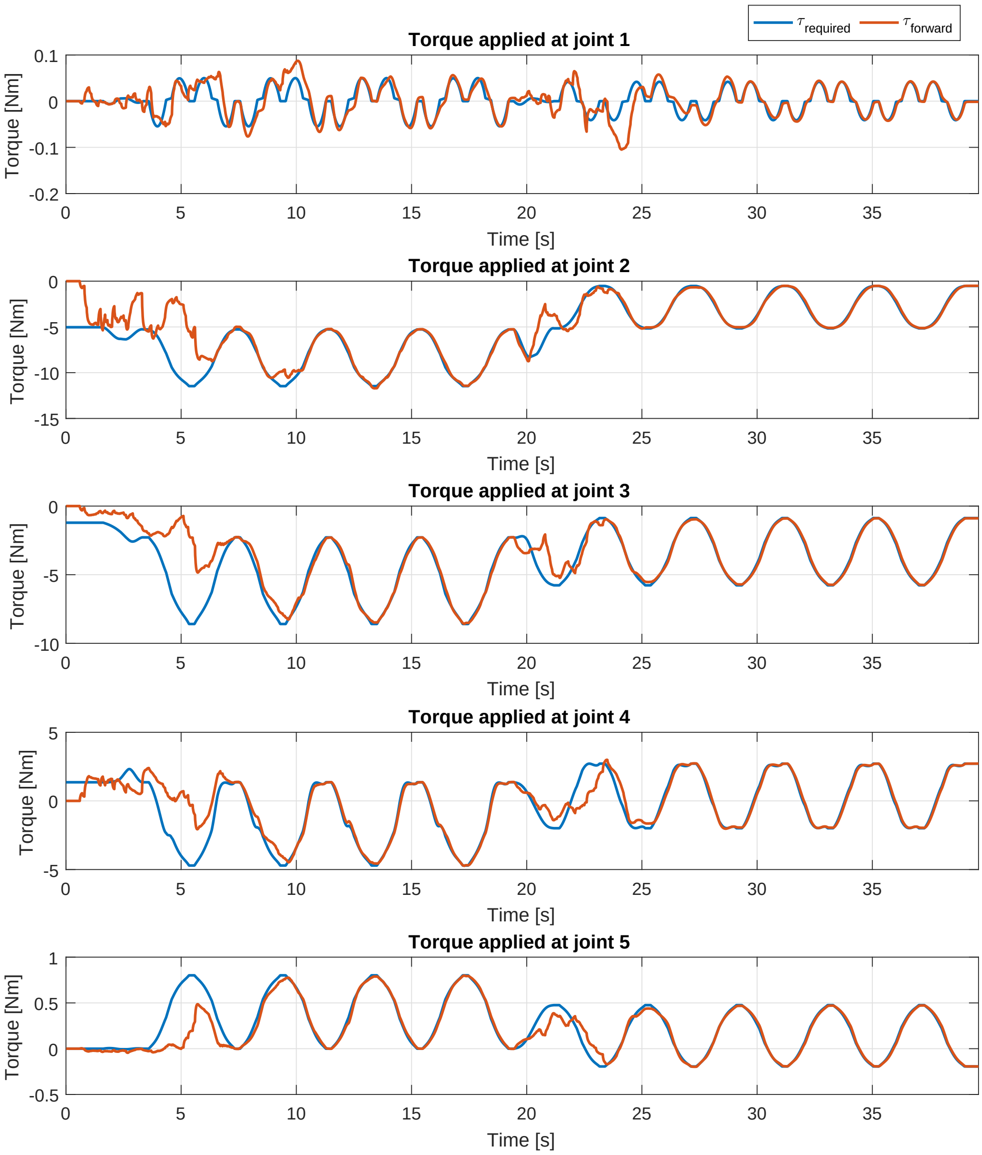


Figure C-2: Plots related to Subsection 5-2-3: Torques required for perfect tracking compared to the torques applied through the control action specified by GPR predictions

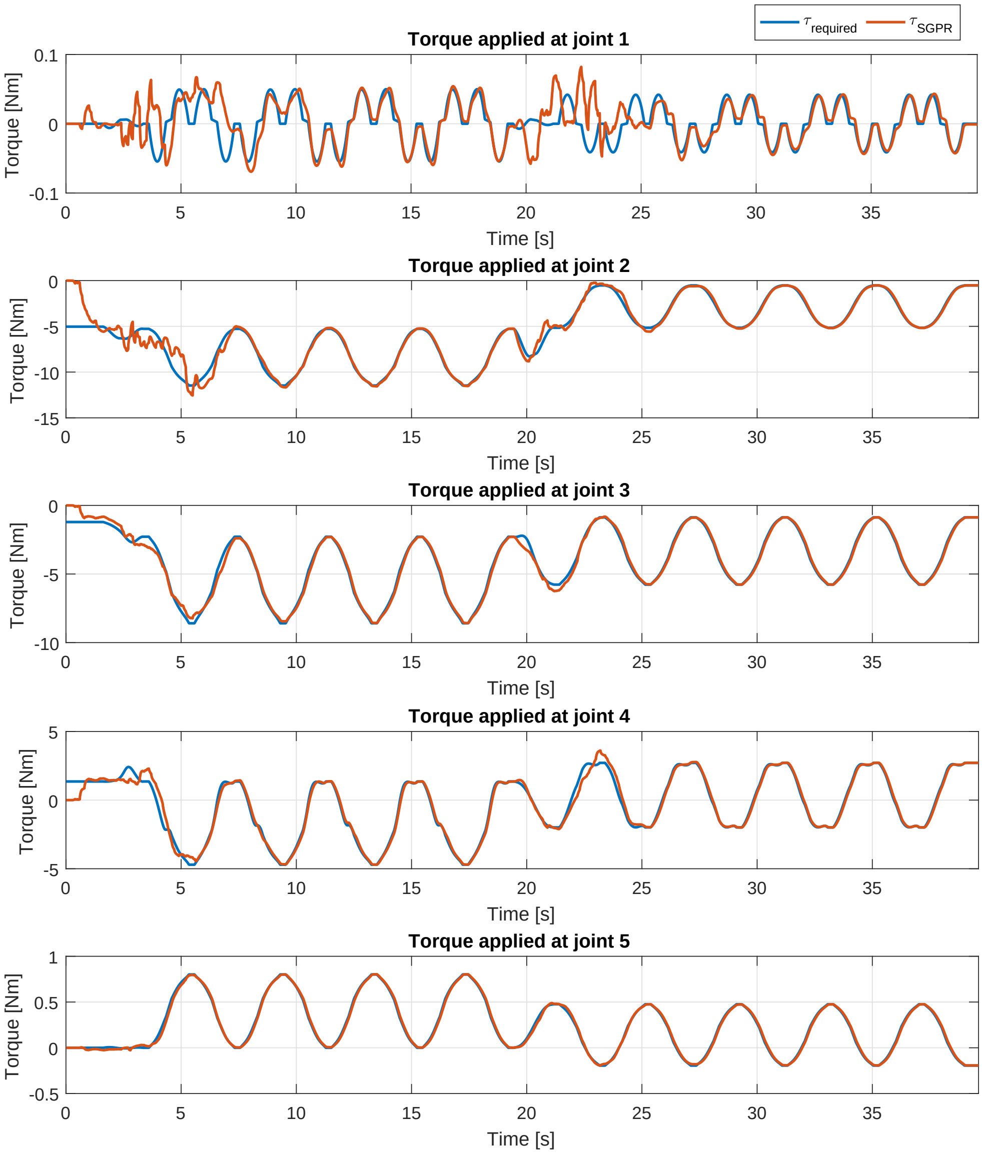


Figure C-3: Plots related to Subsection 5-2-4: Torques required for perfect tracking compared to the torques applied through the control action specified by GPR predictions with EKF parameter identification

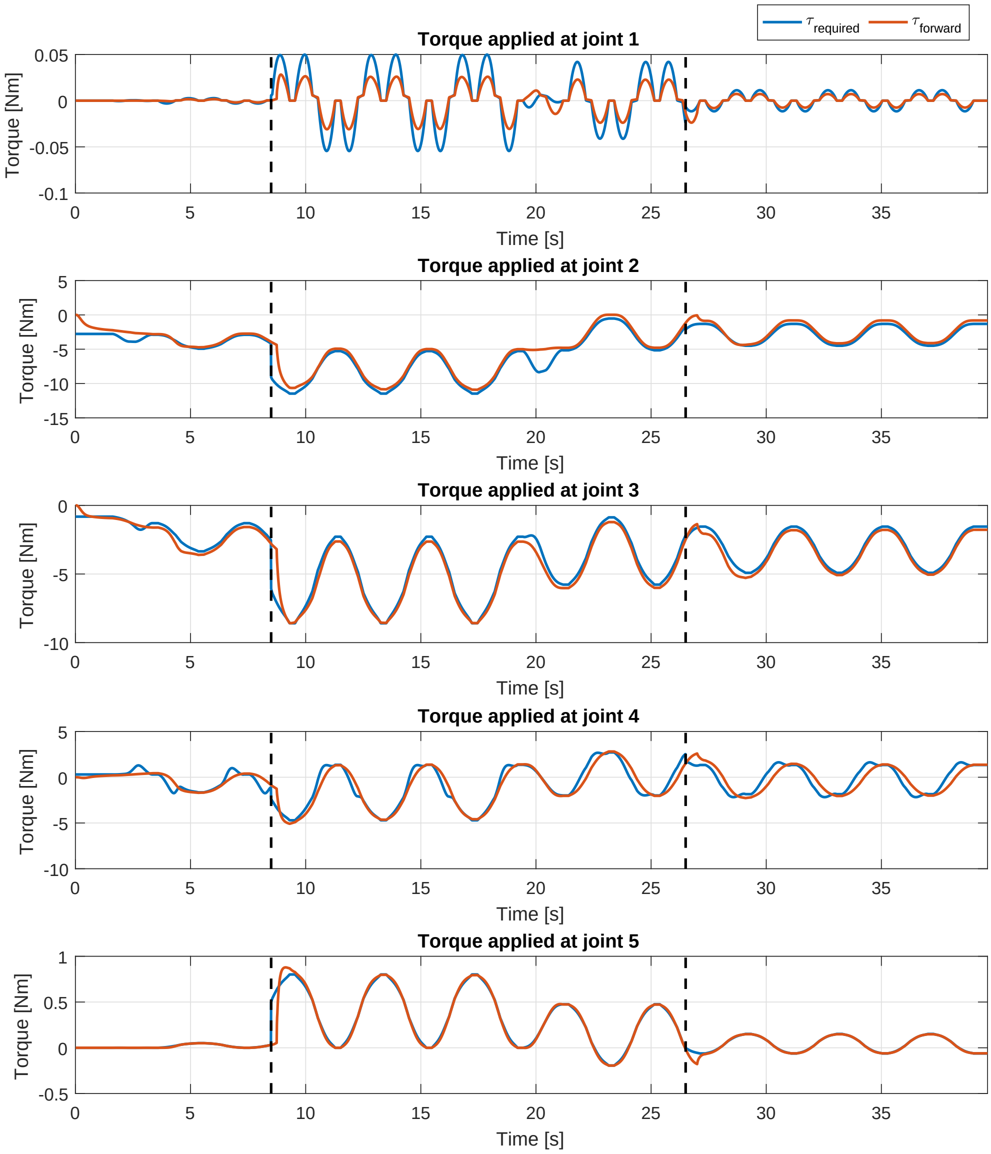


Figure C-4: Plots related to Section 5-3: Torques required for perfect tracking compared to the torques applied through the control action based on EKF predictions. The Black dashed lines indicate instantaneous system changes.

Bibliography

[1] International Federation of Robotics, “Executive Summary World Robotics 2016 Industrial Robots.” http://www.ifr.org/industrial-robots/statistics/, 2016.

[2] R. McCutcheon, R. Pethick, B. Bono, and M. Burak, “The New hire: How a New Generation of Robots is Transforming Manufacturing.” http://www.pwc.com/us/en/industrial-products/publications/next-manufacturing-robotics.html, 2014.

[3] B. Brumson, “Unique Robotic Applications.” http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Unique-Robotic-Applications/content_id/2572, 2011.

[4] T. M. Anandan, “The Business of Automation, Betting on Robots.” http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/The-Business-of-Automation-Betting-on-Robots/content_id/6076, 2016.

[5] M. Bélanger-Barrette, “Collaborative Robot Ebook: Sixth Edition.” http://blog.robotiq.com/collaborative-robot-ebook, 2015.

[6] T. M. Anandan, “The Realm of Collaborative Robots: Empowering Us in Many Forms.” http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/The-Realm-of-Collaborative-Robots-Empowering-Us-in-Many-Forms/content_id/4854, 2014.

[7] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control.* New York: John Wiley and Sons, 2006.

[8] D. Nguyen-Tuong, M. Seeger, and J. Peters, “Model Learning with Local Gaussian Process Regression,” *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[9] G. A. D. Lopes, “Control Methods for Robotics - SC4240TU, Lecture slides.” http://www.dcsc.tudelft.nl/ sc4240tu/index.html, 2015.

[10] R. Ham, T. Sugar, B. Vanderborght, K. Hollander, and D. Lefeber, “Compliant Actuator Designs,” *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 81–94, 2009.

[11] B. Vanderborght, B. Verrelst, R. V. Ham, J. Naudet, J. Vermeulen, D. Lefeber, and F. Daerden, “LUCY, a Bipedal Walking Robot with Pneumatic Artificial Muscles,” *Mechatronics*, 2004.

[12] K. R. Corporation, “LBR iiwa.” https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa, 2017.

[13] J. Schreiter, D. Nguyen-Tuong, and M. Toussaint, “Efficient Sparsification for Gaussian Process Regression,” *Neurocomputing*, vol. 192, pp. 29–37, 2016.

[14] A. S. Polydoros and L. Nalpantidis, “A Reservoir Computing Approach for Learning Forward Dynamics of Industrial Manipulators,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 612–618, 2016.

[15] M. Haruno, D. M. Wolpert, and M. Kawato, “Mosaic Model for Sensorimotor Learning and Control.,” *Neural computation*, vol. 13, pp. 2201–2220, 2001.

[16] G. Petkos, M. Toussaint, and S. Vijayakumar, “Learning Multiple Models of Non-Linear Dynamics for Control under Varying Contexts,” in *International Conference on Artificial Neural Networks*, 2006.

[17] V. Joukov, V. Bonnet, G. Venture, and D. Kulić, “Constrained Dynamic Parameter Estimation using the Extended Kalman Filter,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3654–3659, 2015.

[18] D. Ruppert, M. Wand, and R. Carroll, *Semiparametric Regression.* New York: Cambridge University Press, 2003.

[19] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, “Operational Space Control: A Theoretical and Empirical Comparison,” *International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[20] A. De Luca, “Robotics 2: Dynamic model of robots: Analysis, properties, extensions, parametrization, identification, uses.” http://www.diag.uniroma1.it/ deluca/rob2_en.php, 2015.

[21] J. Peters and D. Nguyen-Tuong, “Model Learning for Robot Control : A Survey,” *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[22] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach.* New York: Cambridge University Press, 1 ed., 2007.

[23] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: MIT-Press, 2002.

[24] S. Vijayakumar and S. Schaal, “Locally Weighted Projection Regression: An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space,” in *International Conference on Machine Learning*, pp. 1079–1086, 2000.

[25] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning.* Massachusetts Institute of Technology: MIT-Press, 2006.

[26] D. Nguyen-Tuong and J. Peters, "Using Model Knowledge for Learning Inverse Dynamics," in *IEEE International Conference on Intelligent Robotics and Automation*, pp. 2677–2682, 2010.

[27] Z. Xue and H. Schwartz, "A Comparison of Several Nonlinear Filters for Mobile Robot Pose Estimation," in *IEEE International Conference on Mechatronics and Automation*, vol. 1, pp. 1087–1094, 2013.

[28] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Computed Torque Control with Nonparametric Regression Models," in *American Control Conference*, pp. 212–217, 2008.

[29] J. Quiñonero-Candela and C. E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[30] D. Nguyen-Tuong and J. Peters, "Local Gaussian Process Regression for Real-time Model-based Robot Control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 380–385, 2008.

[31] K. S. Narendra and J. Balakrishnan, "Adaptive Control Using Multiple Models," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 171–187, 1997.

[32] E. Guizzo and E. Ackerman, "How Rethink Robotics Built Its New Baxter Robot Worker." http://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker, 2012.

[33] M. Florek-Jasińska, "YouBot Detailed Specifications." http://www.youbot-store.com/wiki/index.php/YouBot_Detailed_Specifications, 2015.

[34] MATLAB, *version 9.1.0 (R2016b)*. Natick, Massachusetts: The MathWorks Inc., 2016.

[35] P. Corke, "Robotics toolbox version 9.10." http://petercorke.com/wordpress/toolboxes, 2015.

[36] C. E. Rasmussen and H. Nickisch, "GPML toolbox version 4.0." http://gaussianprocess.org/gpml/code/matlab/release/oldcode.html, 2016.

[37] S. Särkkä, J. Hartikainen, and S. Arno, "EKF/UKF toolbox version 1.3." http://becs.aalto.fi/en/research/bayes/ekfukf/, 2011.

[38] M. S. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding Probabilistic Sparse Gaussian Process Approximations," *arXiv Electronic Journal*, pp. 1–18, 2016.

Glossary

List of Acronyms

DCSC	Delft Center for Systems and Control
DH	Denavit-Hartenberg
DoF	degree-of-freedom
EKF	extended Kalman filter
GP	Gaussian process
GPR	Gaussian process regression
PD	proportional-derivative
RBD	rigid body dynamics
RMSE	root-mean-square error
VFE	variational free energy

List of Symbols

$\boldsymbol{\varepsilon}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$	Time-varying (non)linear (non-)conservative forces
$\boldsymbol{\omega}$	Angular velocity
$\boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$	State dependent regressor
$\boldsymbol{b}_l$	Lower sigmoid bound
$\boldsymbol{b}_s$	Sigmoid slope
$\boldsymbol{b}_u$	Upper sigmoid bound
$\boldsymbol{c}$	Center of mass
$\boldsymbol{C}(t, \boldsymbol{q}, \dot{\boldsymbol{q}})$	Time-varying Coriolis and centripetal forces

$\boldsymbol{e}$	Angular joint position error
$\boldsymbol{e}^{\boldsymbol{\tau}}$	Torque prediction error
$\dot{\boldsymbol{e}}$	Angular joint velocity error
$\boldsymbol{G}(t, \boldsymbol{q})$	Time-varying gravity forces
$\boldsymbol{I}$	Moments of inertia matrix
$\mathbb{I}$	Identity matrix
$\boldsymbol{J}(\boldsymbol{q})$	Jacobian
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{K}_p$	Proportional gain
$\boldsymbol{K}_v$	Derivative gain
$\boldsymbol{K}_{*,*}$	Test data auto-covariance
$\boldsymbol{K}_{\boldsymbol{x},*}$	Cross-covariance between training and test data
$\boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}}$	Training data auto-covariance
$\mathcal{K}$	Kinetic energy
$\boldsymbol{M}$	Process noise covariance matrix
$\boldsymbol{M}(t, \boldsymbol{q})$	Time-varying mass matrix
$\boldsymbol{N}$	Measurement noise covariance matrix
$\boldsymbol{P}$	State covariance matrix
$\mathcal{P}$	Potential energy
$\boldsymbol{q}$	Angular joint position
$\dot{\boldsymbol{q}}$	Angular joint velocity
$\ddot{\boldsymbol{q}}$	Angular joint acceleration
$\boldsymbol{r}$	Cartesian joint position w.r.t. base frame
$\dot{\boldsymbol{r}}$	Cartesian joint velocity w.r.t. base frame
$\ddot{\boldsymbol{r}}$	Cartesian joint acceleration w.r.t. base frame
$\boldsymbol{v}$	Linear velocity
$\boldsymbol{w}$	Linear parameter vector
$\boldsymbol{X}$	Training input data set
$\boldsymbol{x}$	State
k	Time step index
m	Mass
s	Sampling rate
t	Time

Index