

Diploma Thesis



# **Force Estimation in Robotic Manipulators: Modeling, Simulation and Experiments**

**The UR5 Manipulator as a Case Study**

Katharina Kufieta

January 29, 2014

Supervisor:

*Prof. Jan Tommy Gravdahl*

Department of Engineering Cybernetics  
NTNU Norwegian University of Science and Technology



## **Abstract**

The main objective of this work is to set up and test an experimental test stand, including an industrial manipulator, a force sensor and an accelerometer, to perform research on force estimation without force sensors. This includes accurately modeling the industrial manipulator UR5 from Universal Robots, which is used as a case study. Special attention in the modeling is paid to the non-uniform mass distribution in the links of the manipulator. Simulations and experiments of the new model show an improved accuracy as compared to existing dynamic models. Experimental results with the accelerometer, which is attached to the manipulator, indicate its value to validate and improve the manipulator model. A simple force estimation method based on the assumption of a constant current-torque relationship is used to test the experimental setup. The difference between the estimated torques, based on the measured currents, and the expected torques in the joints indicates the presence and magnitude of the contact forces. The experimental results emphasize the importance of friction modeling and the potential of the simple current-torque model for force estimation.

## **Zusammenfassung**

Das Ziel dieser Arbeit ist die Entwicklung einer Kraftschätzung ohne Kraftsensoren und damit einhergehend die Entwicklung eines Versuchsaufbaus mit einem industriellen Manipulator, Kraftsensor und Beschleunigungssensor. Ein wichtiger Bestandteil ist dabei die sorgfältige Modellbildung des industriellen Manipulators UR5 von Universal Robots, der als Fallstudie verwendet wird. Besondere Beachtung bei der Modellierung wird dabei der ungleichmäßigen Massenverteilung in den Gliedern des Roboters zuteil. Simulationen und Experimente mit dem so neu gebildeten Modell zeigen eine verbesserte Genauigkeit im Vergleich zu anderen existierenden Modellen. Die Verwendung eines Beschleunigungssensors, der am Manipulator befestigt ist, deutet auf seinen Nutzwert für die Modellverifizierung und -verbesserung hin. Eine einfache Methode zur Kraftschätzung, basierend auf der Annahme eines konstanten Strom-Moment Zusammenhangs, wird verwendet, um den Teststand zu prüfen. Die Differenz zwischen den geschätzten Momenten, errechnet aus den gemessenen Strömen, und den zu erwartenden Momenten in den Motoren weist auf eine anliegende Kraft hin. Die experimentellen Ergebnisse weisen auf die Bedeutung der Reibungsmodellierung und das Potential eines einfachen Strom-Moment Modells hin.



# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. State of the Art . . . . .	2
1.1.1. Force Estimation on Robotic Manipulators . . . . .	3
1.1.2. Employment of Accelerometers on Robotic Manipulators . . . . .	4
1.1.3. Summary . . . . .	5
1.2. Goals of the Thesis . . . . .	6
1.3. Outline . . . . .	7
<b>2. Principles of Modeling and Control of Industrial Manipulators</b>	<b>9</b>
2.1. Introduction to Industrial Manipulators . . . . .	9
2.2. Modeling of Industrial Manipulators . . . . .	10
2.2.1. Rigid Motions and Homogeneous Transformations . . . . .	11
2.2.2. Forward Kinematics . . . . .	14
2.2.3. Velocity Kinematics . . . . .	18
2.2.4. Inertia Tensors . . . . .	23
2.2.5. Dynamics . . . . .	28
2.3. Control of Industrial Manipulators . . . . .	30
2.3.1. Multivariable Control . . . . .	32
2.3.2. Closed Control Architecture . . . . .	34
2.4. Summary . . . . .	36
<b>3. Principles of Force Estimation</b>	<b>39</b>
3.1. Singularities . . . . .	39
3.2. Force Estimation . . . . .	40
3.2.1. Coulomb Friction Model and Viscous Friction Model . . . . .	41
3.3. Summary . . . . .	43
<b>4. Laboratory Equipment: Manipulator, Force Sensor and Accelerometer</b>	<b>45</b>
4.1. The UR5 Manipulator . . . . .	45
4.1.1. Control Methods . . . . .	45
4.1.2. Provided Parameters and Measurements . . . . .	47
4.2. ATI Force/Torque Sensor . . . . .	51

4.3.	Xsens Accelerometer . . . . .	52
4.4.	Setup of the Test Stand and Software Implementation . . . . .	53
4.4.1.	Software Requirements and Implementation . . . . .	54
4.4.2.	Program Synchronization . . . . .	57
4.4.3.	Data Visualization and Matlab Processing . . . . .	57
4.4.4.	Software Download . . . . .	58
4.5.	Summary . . . . .	58
<b>5.</b>	<b>UR5 Industrial Manipulator Model</b>	<b>59</b>
5.1.	PD Parameter Estimation for the Closed Loop Model . . . . .	59
5.2.	Motivation for a New Dynamic Model . . . . .	61
5.3.	Modeling of the UR5 Manipulator . . . . .	66
5.3.1.	Forward Kinematics . . . . .	66
5.3.2.	Velocity Kinematics . . . . .	71
5.3.3.	Inertia Tensors . . . . .	72
5.4.	Simulation and Verification of the New UR5 Manipulator Models	80
5.5.	Verification of the End Effector Acceleration . . . . .	87
5.6.	Summary and Conclusion . . . . .	92
5.6.1.	Summary . . . . .	92
5.6.2.	Conclusion . . . . .	92
<b>6.</b>	<b>Force Estimation on the UR5</b>	<b>93</b>
6.1.	Current Torque Relationship . . . . .	93
6.2.	Force Estimation at the End Effector . . . . .	97
6.3.	Summary and Conclusion . . . . .	107
6.3.1.	Summary . . . . .	107
6.3.2.	Conclusion . . . . .	107
<b>7.</b>	<b>Conclusion and Outlook</b>	<b>109</b>
7.1.	Conclusion . . . . .	109
7.2.	Outlook . . . . .	111
<b>A.</b>	<b>Definitions and Notions</b>	<b>113</b>
A.1.	Basic Rotation and Transformation Matrices . . . . .	113
A.2.	Skew Symmetric Matrices . . . . .	114
A.3.	The Two-Argument Arctangent Function . . . . .	115
<b>B.</b>	<b>Software Download</b>	<b>117</b>

<b>C. Parameter Values for the UR5 Dynamic Models</b>	<b>119</b>
C.1. PD Control Parameters . . . . .	119
C.2. Parameters Used for the Inertia Tensor Approximation . . . . .	120
<b>D. Acronyms and List of Symbols</b>	<b>123</b>
<b>Bibliography</b>	<b>133</b>





# 1. Introduction

Force control plays a major role in modern robotics. It is, together with vision-based control, a key technology to enable the integration of robotics in unknown or human environments. Both vision and force controlled manipulators react to obstacles as soon as these are detected, enabling robots to avoid collisions and thus avoiding damage to themselves, the surrounding objects or injuries to nearby humans. Successfully implemented force control can therefore allow a robot to complete more advanced or flexible automation tasks. Examples are tasks such as coring, drilling, grasping, scooping, constructing or robotic assembly with an uncertainty about the position of the assembly parts. Another application for force control is free movement in unknown environments, detection of obstacles and mapping of surroundings. Force control also enables the manipulator to be part of a production line in direct proximity to humans without being contained in robot cages.

Force control has, in the presence of uncertainties, a clear advantage over traditional manipulator control which focuses on precision position control. Precision position control methods expect an environment designed to a similarly precise specification level. The manipulator is programmed to track a time varying joint trajectory specified to accomplish a well-defined task. Unplanned changes in the environment can therefore lead to damage to both the robot and the surrounding objects. If the assignment is changed, a renewed software implementation and a new layout of the robot surroundings is often needed. The resulting costs for those changes can be very high, as the software implementation and the design of the surroundings must satisfy high precision demands. Depending on the type of the assignment, a force controlled approach can be more beneficial in the long run.

Robot force control is a mature research field, its earliest results date back to 1981 [26]. Classical force control tasks are assembly, grinding and deburring, as they involve extensive contact with the environment. Typical force control methods are impedance control, parallel control and hybrid force/position control. Current force control challenges stem from contact with more flexible environments, like in robotized sewing [23]. Commonly, a force/torque (F/T) sensor is mounted either externally or internally at the robot wrist, or

joint torque measurements are used if they are integrated in the manipulator joints.

Force sensors can be very expensive and they introduce complexity into a manipulator's mechanical and electrical design. In order to avoid this overhead, force estimation as a research field gained attention from both manipulator manufacturers and in research communities. Results in force control without force sensors for a 2 Degrees Of Freedom (DOF) manipulator were already available in 1991 [29], where robot joint velocities and accelerations together with a dynamic model were used to perform impedance control without a force sensor.

A literature review of force estimation on robotic manipulators as well as the utilization of accelerometers is presented in the next section.

### 1.1. State of the Art

A UR5 industrial manipulator is used as a case study for force estimation without force sensors in this thesis. A characteristic of this manipulator is a closed control architecture which only gives the user access to program high-level applications such as defining joint trajectories, rather than to designing custom low-level control applications. Closed control architectures are a common problem when researching commercially available industrial manipulators. The following literature review therefore pays special attention to the openness of the underlying manipulator control architectures, and the impact this has on the suggested force estimation methods.

The goal in this study is to utilize the current measurements in the motors of the manipulator to detect contact forces acting on the robot. The UR5 has brushless alternating current (AC) motors and follows hereby the present trend in the design of industrial manipulators. As highlighted by some publications, AC motors appear to be a major source of inherent friction which complicates the force estimation based on current measurements.

A further consideration is to attach an accelerometer to the manipulator and thus obtain more measurements directly related to the kinetic energy of the manipulator. By assuming that the manipulator is an assembly of rigid bodies, the measured acceleration can be assumed to be caused only by the acceleration in the joints. Hence, a literature review about the use of accelerometers on industrial manipulators follows the literature review of force estimation.

### 1.1.1. Force Estimation on Robotic Manipulators

In [28], the authors estimate contact force based on the position control error. The method is based on detuning the low-level joint control loops. Experiments are performed on an ABB FRIDA robot with access to the low-level Proportional-Integral-Derivative (PID) control parameters. The force estimation is affected by how the integral part of the controller is chosen, such that fine tuning is needed to match the estimated force to the measured force. The estimated force shows high disturbances, less so if a priori information about the external force is used, which is only possible if the environment in which the robot acts is partly known prior to the application. Furthermore, full access to the PID control parameters is necessary to perform this method.

A robust force estimation algorithm is proposed in [16] for estimating the 3D contact force acting on a three-link robot manipulator. The algorithm is developed by combining the Extended Kalman filter (EKF) with an adaption law for non-linear stochastic systems with unknown inputs. To test the theory, simulations are performed, where the joint positions and velocities of the three-link manipulator are used as measurements.

End effector force estimation based on noisy actuator torque measurements is attempted by the authors in [33]. Two methods are presented, where the first method uses a filtered dynamic model and a recursive least-squares estimation with exponential forgetting to estimate the force. The second estimation approach is based on the generalized momentum based disturbance observer. Experiments are performed on a 2-DOF manipulator with pneumatic muscle actuators. The end effector forces in  $x$ - and  $y$ -direction are estimated successfully in a range of forces between  $\pm 6$  N, despite of the noise.

Static friction, also called stiction, which is inherent in harmonic drives was given special attention in the study [1]. Friction is difficult to model and therefore a major source of error in force estimation. Manipulator dynamics are used together with motor current feedback to estimate external joint torques, which are transformed into estimated external end effector forces. Friction learning by means of neural networks and adaptive control to tune the parameters of the robot's modeled dynamics are used to adapt to changing conditions during operation of the manipulator. A flaw of the method is that the manipulator has to switch between training mode, where the external force acting on the manipulator is assumed to be zero, and estimation mode, when it is expected that external forces are present. It will therefore not work in entirely unknown environments, when the occurrence of external forces can happen at any time. Experiments were performed on a 2-DOF manipulator with full access to low

level controllers.

In [25], the authors use motor currents together with an accurate system model to estimate external forces for robots with harmonic drive gearing. The approach involves subtracting modeled dynamics from motor torque, assumed to be proportional to motor current, to form the estimated external torque. The estimated torque thus obtained contains significant unmodeled position-dependent friction. Filtering the estimated external torque in the position domain greatly improves the estimates. A flaw of this force estimation method is that it relies on unchanging conditions in the manipulator dynamics. The entire position history of the movement, and where the forces are acting on the manipulator, is known. The force estimation was performed in 2D on a Hirata ARI350 SCARA Robot, where fast sample rates are provided and own implementations of control methods are possible.

Five techniques for sensing forces with a manipulator are compared analytically and experimentally by the authors in [14]. The comparison includes a six-axis wrist F/T sensor, joint torque sensors, link strain gauges, motor current sensors and flexibility modeling. The primary investigation is to explore the challenge of using force sensing equipment on a landed flight mission with the intent of performing tasks such as coring, drilling, grasping and assembly on a foreign planetary surface. In the case of the force estimation based on motor current sensors, a detailed description of the underlying motor is given, which involves the gear ratio, the gear train efficiency and the torque constant. Experiments are performed on a 5-DOF experimental arm with low-level control access.

### 1.1.2. Employment of Accelerometers on Robotic Manipulators

6-Dimensional (D) F/T and 6D acceleration measurements were fused in the study [17] to detect contact forces for the case that the force sensor is permanently integrated between the manipulator and the end effector. The mass of the end effector will cause inertial forces acting on the force sensor when the arm is moving. By knowing the mass and by means of an accelerometer these inertial forces can be calculated and cancelled out from the F/T measurements. Experiments were performed on a Staeubli RX60 industrial manipulator where the original control unit had been replaced by a new one, so low-level joint control as well as higher levels of control was possible.

A similar sensor fusion technique is used in the collection of papers [7–13] that integrates the information of a wrist force sensor, a 3D accelerometer placed at the robot tool and the joint position sensor measurements. The goal

is to rule out inertial forces acting on the force sensor when the arm is moving. Additionally a self-calibrating feature is built in that allows easy integration of the method into any industrial setup. Experiments were performed on two different manipulators: an ABB robot and a Stäubli robot, both with open control system architectures.

The authors [3] use an accelerometer mounted at the end effector to improve the accuracy of the arm angular position, as well as the estimated position of the end effector. Experiments are performed on an ABB IRB4600 robot. The same author presents a method to find the orientation and position of a triaxial accelerometer mounted on a 6-DOF industrial robot in [4].

Low-cost 3D Micro Electro-Mechanical System (MEMS) accelerometers are used in the study [20] to estimate the joint angles of robotic manipulators instead or in addition to the use of shaft encoders. For this, at least one 3D accelerometer is used for each pair of joints and an EKF scheme is used by measuring gravity in the joint frame and relating the joint angles with the measurements using rotation matrices. A flaw of this method is that the estimation breaks down when the rotation axis is parallel to the direction of gravity. Experiments are performed on a Willow Garage PR2 Alpha and a prototype low-cost manipulator, both with open control architecture.

Similar to the publication [20], low cost triaxial accelerometers are used in [22] to estimate the joint angles and velocities of a manipulator. Additionally gyroscopes are used to cover the case when the accelerometer cannot provide useful information in the case that the rotation axis is parallel to the direction of gravity. Experiments are performed on a Willow Garage PR2 Alpha.

Similar to [20] and [22], the authors in [34] use low-cost MEMS accelerometers and gyroscopes to estimate the joint positions of a manipulator. But instead of using only the gravity, it uses a full kinematic model for the acceleration and the angular rate measurements. An EKF is used to estimate the joint positions. Experiments are performed on a test stand representing two consecutive revolute joints.

### **1.1.3. Summary**

Force estimation is often performed on prototype robotic manipulators with full access to the implementation of the controller. In the case that industrial manipulators are used, access to the PID control parameters is given. The open control architecture is often a requirement for the application of the developed force estimation methods. These methods are either based on the joint position and velocity tracking errors, actuator torque measurements or motor current

feedback. Actuator torque measurements do often require torque sensors in the joints. Motor current feedback from harmonic drives carry along high disturbance stemming from dynamic and static friction. Friction modeling is therefore important for the force estimation based on motor currents.

The application of accelerometers on robotic manipulators is mostly used for sensor fusion of acceleration and force measurements in order to subtract torques and forces stemming from inertia caused by the mass of the end effector. Another use of accelerometers on manipulators is for joint position estimation. By using low-cost accelerometers instead of shaft encoders the costs for the production of manipulators can be reduced significantly. This is desirable in order to make robots acquirable for personnel or domestic use.

The approach of using accelerometers to improve force estimation without using force sensors is not yet employed on industrial manipulators.

### 1.2. Goals of the Thesis

One of the main objectives of this thesis is to set up and test an experimental test stand, including the industrial manipulator UR5, an ATI F/T sensor and an Xsens accelerometer. The goal is to record measurements from manipulator, force sensor and accelerometer in realtime while controlling the robot along desired trajectories. The obtained data is then supposed to be processed further in Matlab. Ideally, the test stand is well documented and easy to use for subsequent users.

In order to perform research in force estimation an accurate dynamic model of the industrial manipulator UR5 is required. A model is derived paying special attention to the non-uniform mass distribution in the links of the manipulator. A variety of link modeling techniques are used to compare the performance of the resulting dynamic models with the model as derived in the thesis [21].

Tests are then performed by comparing modeled and measured acceleration of the end effector and by performing a simple force estimation method based on the assumption of a constant current-torque relationship. A deviation of the estimated torques that are based on the measured currents from the expected torques is supposed to indicate the presence and magnitude of forces acting on the manipulator.

The outline of this thesis is presented in the next section.

### **1.3. Outline**

Chapter 2 presents the principles of modeling that are used to derive the dynamic model of the UR5. This includes forward and velocity kinematics, inertia tensors and dynamic equations. Typical control methods used in industrial manipulators are mentioned together with the problems with a closed control architecture. An estimation of the Proportional-Derivative (PD) control parameters is suggested in order to complete the modeling of the manipulator in a closed loop.

Chapter 3 presents the principles of force estimation that are applied in this thesis. Singularities are mentioned before introducing a simple method for force estimation based on motor currents, together with the basics of friction modeling.

Chapter 4 introduces the equipment used for the experimental test stand. It continues with an explanation of the setup of the test stand, the implementation requirements of the devices and the software implementation performed to automate the experiment.

Chapter 5 introduces the model as derived in the thesis [21] and gives a subsequent motivation for the derivation of a new model. Three dynamic models are then derived based on different link modeling techniques. Their performance is compared and a model is chosen for the remainder of the thesis. The modeled acceleration at the end effector is then compared to the measurements of the accelerometer.

Chapter 6 describes the implemented force estimation, which includes finding a current-torque relationship. The estimated forces are then compared to the measurements of the force sensor.

Chapter 7 summarizes and concludes the thesis and gives recommendations for future work.





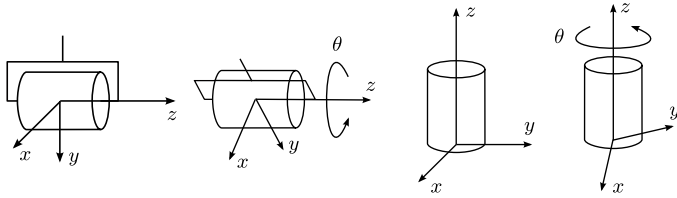
## 2. Principles of Modeling and Control of Industrial Manipulators

After an introduction to the basics and terminology that are frequently used for industrial manipulators, this chapter gives an overview of the principles of modeling and control for manipulators as presented in [26]. The introduction is given in Section 2.1. Section 2.2 presents the derivation of the forward and velocity kinematics, methods to approximate the inertia tensors and the dynamic model equations based on the Euler-Lagrange equations. Common control methods in industrial manipulators for both joint space and task space, the problem of a closed control architecture and examples of the control structure of popular lightweight manipulators are presented and discussed in Section 2.3. Furthermore, a method is presented to estimate the control parameters of a manipulator, when these are unknown and hidden by a closed control architecture. The scope of the presented material is restricted to one of the most common industrial manipulator types, i.e., a manipulator with six rotary joints and a total of six DOF.

### 2.1. Introduction to Industrial Manipulators

Industrial Manipulators are robots with a mechanical arm operating under computer control. They are composed of links that are connected by joints to form a kinematic chain. The manipulator that is considered in this thesis has solely rotary, also called revolute, joints. Each represents the interconnection between two links. The axis of rotation of a revolute joint, denoted by  $z_i$ , is the interconnection of links  $l_i$  and  $l_{i+1}$ . The joint variables, denoted by  $q_i$ , represent the relative displacement between adjacent links. As the joints in this study are revolute, it holds in this special case that  $q_i = \theta_i$ ,  $\forall i \in \{1, \dots, n\}$ , where  $\theta_i$  denotes a relative rotation. Both notations are used throughout the thesis. A symbolic representation of this displacement for revolute joints is shown in Figure 2.1.

The specification of the location of every point on the manipulator is called the configuration of the robot. The set of all configurations is called the config-



**Figure 2.1.:** Symbolic representation of revolute joints [26]. Each joint allows a single degree of freedom of motion between adjacent links of the manipulator. The revolute joint produces a relative rotation between adjacent links.

uration space. As the base of manipulators is commonly fixed and the links are assumed to be rigid, the configuration is defined by knowing the values for the joint variables. These are often gathered into a vector  $\mathbf{q} = [q_1, \dots, q_n]^T$ . The joint velocities are then  $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_n]^T$ .

The following section gives an introduction to the modeling of industrial manipulators.

## 2.2. Modeling of Industrial Manipulators

The derivation of the forward and velocity kinematics and the dynamic equations for typical industrial manipulators is well established, documented and tested [5, 24, 26]. The modeling of manipulators is therefore often simplified to step-by-step procedures. Rigid motions and homogeneous transformations are defined to represent the positions and orientations of rigid objects and the rotation and translation between assigned coordinate frames. A method called the Denavit-Hartenberg (DH) convention was developed to standardize the assignment of coordinate frames to joints and links of manipulators and to create homogeneous transformation matrices. By using those matrices it is easy to derive the forward kinematics. In order to derive the velocity kinematics a Jacobian is defined specifically for robotic manipulators, also called the manipulator Jacobian. By using this Jacobian and either the Euler-Lagrange equations or the Newton-Euler formulation it is possible to derive the dynamic equations for the manipulator.

Despite all those simplifications it can be a challenging task to model industrial manipulators as specific knowledge about the manipulator parameters is needed. These parameters include the dimensions of the robot, the weight and

the inertia tensors of the joints and links and motor specifications of the joints. Depending on the manufacturer those parameters can be either openly available, partly accessible or even kept secret, and they are difficult to estimate. Eventually the resulting dynamic equations consist of large matrices which cannot be verified analytically, but rather through extensive simulations and experiments. The possibility to verify the manipulator model depends again on the number of sensors and the amount of measurements that are available, which varies greatly between manufacturers and manipulator types.

The following section introduces the notion of rigid motions and homogeneous transformations, which are essential to derive the forward kinematics.

### 2.2.1. Rigid Motions and Homogeneous Transformations

Rigid motions and homogeneous transformations are used to describe the relative positions and orientations between the coordinate systems that are assigned to each joint and its respective link. Homogeneous transformations combine the operations of rotation and translation into a single matrix multiplication which is commonly used to derive the forward kinematic equations of rigid manipulators and to perform coordinate transformations.

Rigid motions are defined to be an ordered pair  $(\mathbf{d}, \mathbf{R})$  where  $\mathbf{d} \in \mathbb{R}^3$  is a translation vector and  $\mathbf{R} \in SO(3)$  is a rotation matrix of the Special Orthogonal group of order three [26]. For any  $\mathbf{R} \in SO(n)$  the property holds that  $\mathbf{R}^T = \mathbf{R}^{-1} \in SO(n)$  and  $\det \mathbf{R} = 1$ . Rotation matrices can be used to represent the orientation of one coordinate frame with respect to another as well as to transform the coordinates of a point from one frame to another. Successive rotations such as a rotational transformation of a frame  $o_i x_i y_i z_i$  to a frame  $o_j x_j y_j z_j$  and further to frame  $o_k x_k y_k z_k$  can be obtained by

$$\mathbf{R}_k^i = \mathbf{R}_j^i \mathbf{R}_k^j \quad (2.1)$$

A vector pointing to a point  $p$  in frame  $o_i x_i y_i z_i$  is denoted by  $\mathbf{p}^i$ .

Arbitrary rotations can be represented by using only three independent quantities. This property is used in common rotation representations such as the Euler-angle representation, the roll-pitch-yaw representation, and the axis/angle representation. Both the Euler-angle representation and the axis/angle representation are used in this thesis and are presented in the following section, followed by an explanation of the homogeneous transformation.

### Euler Angle Representation

The orientation of a frame  $o_j x_j y_j z_j$  relative to a frame  $o_i x_i y_i z_i$  can be specified by a vector of three angles  $\phi = [\varphi, \vartheta, \psi]^T$ , known as Euler angles. A rotation matrix can be obtained by three consecutive rotations, first about the current  $z$ -axis by the angle  $\varphi$ , next about the current  $y'$ -axis by the angle  $\vartheta$  and finally about the current  $z''$ -axis by the angle  $\psi$ . By using the basic rotation matrices as described in Appendix A.1 the resulting rotational transformation can be generated as the product

$$\begin{aligned}
 \mathbf{R}_{ZYZ} &= \mathbf{R}(\phi) = \mathbf{R}_z(\varphi) \mathbf{R}_{y'}(\vartheta) \mathbf{R}_{z''}(\psi) \\
 &= \begin{bmatrix} c_\varphi & -s_\varphi & 0 \\ s_\varphi & c_\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\vartheta & 0 & s_\vartheta \\ 0 & 1 & 0 \\ -s_\vartheta & 0 & c_\vartheta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2) \\
 &= \begin{bmatrix} c_\varphi c_\vartheta c_\psi - s_\varphi s_\psi & -c_\varphi c_\vartheta s_\psi - s_\varphi c_\psi & c_\varphi s_\vartheta & c_\psi s_\vartheta \\ s_\varphi c_\vartheta c_\psi + c_\varphi s_\psi & -s_\varphi c_\vartheta s_\psi + c_\varphi c_\psi & s_\varphi s_\vartheta & s_\psi s_\vartheta \\ & -s_\vartheta c_\psi & & s_\vartheta s_\psi & c_\vartheta & c_\psi \end{bmatrix}
 \end{aligned}$$

The matrix  $\mathbf{R}_{ZYZ}$  is called the ZYZ-Euler angle transformation.

In order to obtain the analytical Jacobian in Section 2.2.3 it is necessary to determine the Euler angles  $\varphi$ ,  $\vartheta$  and  $\psi$  that satisfy Equation (2.2). The following method shows how to derive the Euler angles out of a general rotation matrix  $\mathbf{R}$  with

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.3)$$

There are two cases that can be treated separately, depending on the entries  $r_{13}$  and  $r_{23}$ .

**Case 1.** ( $r_{13} \neq 0 \oplus r_{23} \neq 0$ ) or ( $r_{13} \neq 0 \wedge r_{23} \neq 0$ ).

A possible solution for  $\vartheta$  is

$$\vartheta = \text{Atan2} \left( r_{33}, \sqrt{1 - r_{33}^2} \right) \quad (2.4)$$

where the function  $\text{Atan2}$  is the two-argument arctangent function defined in Appendix A.3. For  $\varphi$  and  $\psi$  it holds that

$$\begin{aligned}
 \varphi &= \text{Atan2}(r_{13}, r_{23}) \\
 \psi &= \text{Atan2}(-r_{31}, r_{32}) \quad (2.5)
 \end{aligned}$$

**Case 2.**  $r_{13} = r_{23} = 0$ .

If  $r_{33} = 1$  the sum  $\varphi + \psi$  can be determined as

$$\varphi + \psi = \text{Atan2}(r_{11}, r_{21}) = \text{Atan2}(r_{11}, -r_{12}) \quad (2.6)$$

If  $r_{33} = -1$  the difference  $\varphi - \psi$  is

$$\varphi - \psi = \text{Atan2}(-r_{11}, -r_{12}) \quad (2.7)$$

Since only the sum  $\varphi + \psi$  or the difference  $\varphi - \psi$  can be determined in this case, there are an infinite number of solutions. To find a solution,  $\varphi = 0$  can be chosen by convention.

### Axis/Angle Representation

The axis/angle representation describes the rotation about an arbitrary axis in space. A unit vector  $\mathbf{l} = [l_x, l_y, l_z]^T$ , which is defining an axis expressed in a frame  $o_i x_i y_i z_i$  together with a rotation of  $\lambda$  about this axis are used to derive the rotation matrix  $\mathbf{R}_{\mathbf{l}, \lambda}$ . The result is

$$\mathbf{R}_{\mathbf{l}, \lambda} = \begin{bmatrix} l_x^2 v_\lambda + c_\lambda & l_x l_y v_\lambda - l_z s_\lambda & l_x l_z v_\lambda + l_y s_\lambda \\ l_x l_y v_\lambda + l_z s_\lambda & l_y^2 v_\lambda + c_\lambda & l_y l_z v_\lambda - l_x s_\lambda \\ l_x l_z v_\lambda - l_y s_\lambda & l_y l_z v_\lambda + l_x s_\lambda & l_z^2 v_\lambda + c_\lambda \end{bmatrix} \quad (2.8)$$

The inverse extraction of the axis  $\mathbf{l}$  and angle  $\lambda$  from a general rotation matrix, written as in Equation (2.3), is given by the expressions

$$\lambda = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \quad (2.9)$$

$$\mathbf{l} = \frac{1}{2 \sin \lambda} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (2.10)$$

### Homogeneous Transformation

Homogeneous transformations simplify the handling of long sequences of rigid motions, as it reduces the composition of rigid motions to matrix multiplication. A homogeneous transformation matrix  $\mathbf{H} \in \mathbb{R}^{4 \times 4}$  has the form of

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{R} \in SO(3), \quad \mathbf{d} \in \mathbb{R}^3 \quad (2.11)$$

Using the fact that  $\mathbf{R}$  is orthogonal the inverse of the homogeneous transformation matrix is simply

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.12)$$

To calculate subsequent transformations, the homogeneous transformation matrices must be multiplied, according to

$$\mathbf{H}_j^i = \mathbf{H}_{i+1}^i \cdots \mathbf{H}_j^{j-1} \quad (2.13)$$

The rotational parts are then given by

$$\mathbf{R}_j^i = \mathbf{R}_{i+1}^i \cdots \mathbf{R}_j^{j-1} \quad (2.14)$$

and the translation vectors are given by

$$\mathbf{d}_j^i = \mathbf{d}_{j-1}^i + \mathbf{R}_{j-1}^i \mathbf{d}_j^{j-1} \quad (2.15)$$

A set of basic homogeneous transformations is given in Equation (A.3).

Building upon the definitions of rigid motions and homogeneous transformation, the forward kinematics can be calculated, as shown in the next section.

### 2.2.2. Forward Kinematics

Kinematics describe the motion of the manipulator without consideration of the forces and torques causing the motion. Forward kinematics are used to determine the position and orientation of the end effector by using the state of the joint variables. In order to derive the forward kinematics a coordinate system has to be assigned to each joint and the link that is attached to it. The DH convention and a few other definitions help to assign the coordinate systems in a standardized way and are presented in the following section.

#### General Definitions

A robot manipulator with  $n$  joints will have  $n + 1$  links where the numbering for the joints is  $j_i : i \in \{1, \dots, n\}$  and for the links it is  $l_i : i \in \{0, \dots, n\}$ , starting from the base where the manipulator is grounded. By this convention, joint  $j_i$  connects link  $l_{i-1}$  to link  $l_i$ .

The location of joint  $j_i$  is considered to be fixed with respect to link  $l_{i-1}$ . A coordinate frame  $o_i x_i y_i z_i$  is rigidly attached to link  $l_i$ . When joint  $j_i$  is actuated,

link  $l_i$  and its attached frame  $o_i x_i y_i z_i$  move. A joint variable  $q_i = \theta_i$  is denoted with the  $i^{\text{th}}$  joint.

The frame  $o_0 x_0 y_0 z_0$ , which is attached to the robot base, is referred to as the base frame or inertial frame.

Homogeneous transformations are crucial to describe the kinematics.  $\mathbf{A}_i$  is a homogeneous transformation matrix that gives the position and orientation of  $o_i x_i y_i z_i$  with respect to  $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$  and is therefore of the form

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{R}_i^{i-1} & \mathbf{o}_i^{i-1} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.16)$$

$\mathbf{T}_j^i$  is a homogeneous transformation matrix that expresses the position and orientation of  $o_j x_j y_j z_j$  with respect to  $o_i x_i y_i z_i$ .  $\mathbf{T}_j^i$  is derived by multiplication of the transformation matrices  $\mathbf{A}_i$  by

$$\mathbf{T}_j^i = \begin{cases} \mathbf{A}_{i+1} \mathbf{A}_{i+2} \cdots \mathbf{A}_{j-1} \mathbf{A}_j, & \text{if } i < j \\ \mathbf{I}, & \text{if } i = j \\ (\mathbf{T}_j^i)^{-1}, & \text{if } i > j \end{cases} \quad (2.17)$$

The position and orientation of the end effector in the inertial frame are then given by

$$\mathbf{T}_n^0 = \mathbf{A}_1(q_1) \cdots \mathbf{A}_n(q_n) = \begin{bmatrix} \mathbf{R}_n^0 & \mathbf{o}_n^0 \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.18)$$

Note that the translation vector is  $\mathbf{d}_n^0 = \mathbf{o}_n^0$  which is often used when the transformation matrices affect coordinate systems rather than general objects.

### Denavit-Hartenberg Convention

Homogeneous transformations are used to describe the relationship between the coordinate frames  $o_i x_i y_i z_i$ , for  $i \in \{0, \dots, n\}$ , that are assigned to the links and joints of the robot. In order to define a rigid motion six parameters are generally needed: three parameters to define the rotation and three parameters to define the translation. The DH convention reduces the number of parameters that are needed to define a homogeneous transformation from six to four by exploiting the common manipulator geometry and by making a clever choice of the origin and the coordinate axes. The DH convention is a standard procedure used in the derivation of the forward kinematics of industrial manipulators.

In the DH convention each homogeneous transformation  $\mathbf{A}_i$  is a product of four basic transformations

$$\begin{aligned}
 \mathbf{A}_i &= \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &\quad \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ics_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_ics_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

where the four parameters  $\theta_i$ ,  $a_i$ ,  $d_i$ ,  $\alpha_i$  are the joint angle, link length, link offset and link twist of joint  $j_i$  and link  $l_i$ . Since  $\mathbf{A}_i$  is a function of  $\theta_i$  the other three parameters are constant.

According to the DH convention the coordinate frames  $o_i x_i y_i z_i$  that are assigned to the joint  $j_i$  and link  $l_i$  must have the following attributes:

1. The axis  $z_i$  is the axis of revolution of joint  $j_{i+1}$ .
2. The axis  $x_i$  is perpendicular to the axis  $z_{i-1}$  and  $z_i$ .
3. The axis  $x_i$  intersects the axis  $z_{i-1}$ .
4. All frames are right-handed.

After setting up the coordinate frames a table with DH parameters  $a_i$ ,  $d_i$ ,  $\alpha_i$  and  $\theta_i$ ,  $\forall i \in \{1, \dots, n\}$ , has to be established according to the following rules:

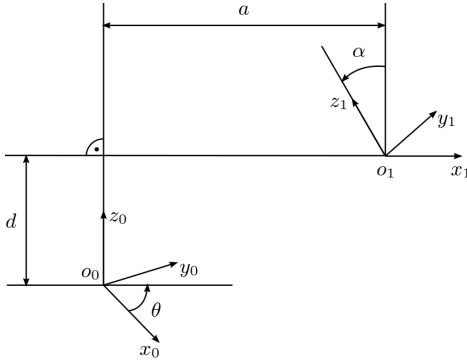
$a_i$  = distance along  $x_i$  from the intersection of the  $x_i$  and  $z_{i-1}$  axes to  $o_i$ .

$d_i$  = distance along  $z_{i-1}$  from  $o_{i-1}$  to the intersection of the  $x_i$  and  $z_{i-1}$  axes.

$\alpha_i$  = the angle from  $z_{i-1}$  to  $z_i$  measured about  $x_i$ .

$\theta_i$  = the angle from  $x_{i-1}$  to  $x_i$  measured about  $z_{i-1}$ . As all the joints in the UR5 are revolute,  $\theta_i$  is a variable.





**Figure 2.2.:** DH parameter derivation. The four parameters are the joint angle  $\theta$ , the link length  $a$ , the link offset  $d$ , and the link twist  $\alpha$ . They are obtained by following the DH convention [26]. Note that for revolute joints only  $\theta$  is a variable, whereas  $d$ ,  $a$  and  $\alpha$  are constants.

Figure 2.2 visualizes the derivation of the DH parameters for two exemplary coordinate frames.

A total of  $n$  homogeneous transformation matrices, namely  $\mathbf{A}_1, \dots, \mathbf{A}_n$ , are obtained by substituting the obtained parameters into Equation (2.19) and will be used to derive the forward kinematics.

## Forward Kinematics

To obtain the forward kinematics the above derived homogeneous transformation matrices  $\mathbf{A}_1, \dots, \mathbf{A}_n$  have to be inserted in Equation (2.18). The resulting transformation matrix describes the transformation from the base to the end effector of the manipulator as a function of the joint variables  $q_i$ , which concludes the forward kinematics problem.

The following section deals with the derivation of the velocity kinematics, including the derivation of the geometric and analytical Jacobian and the Jacobian for an arbitrary point on the manipulator.

### 2.2.3. Velocity Kinematics

Velocity kinematics relate the linear and angular velocities of the end effector to the joint velocities. To obtain them, the forward kinematic equations are used as a basis. The forward kinematics define a function between the end effector position and orientation and the joint positions. The velocity relationships are then determined by the Jacobian of this function.

The manipulator Jacobian  $\mathbf{J} \in \mathbb{R}^{6 \times n}$  is used not only for the derivation of the velocity kinematics, but also in many tasks in robotic manipulation, e.g., planning and execution of smooth trajectories, determination of singular configurations, derivation of the dynamic equations of motion, and the transformation of forces and torques from the end effector to the manipulator joints.

It represents the instantaneous transformation between the joint velocities  $\dot{\mathbf{q}} \in \mathbb{R}^n$  and the linear and angular velocities of the end effector, i.e.,  $\boldsymbol{\xi} \in \mathbb{R}^6$  in the case of a non-minimalistic representation of the orientation of the end effector or  $\mathbf{X} \in \mathbb{R}^6$  for the minimalistic counterpart.  $\mathbf{J}$  is furthermore labeled as the geometric Jacobian  $\mathbf{J}_g \in \mathbb{R}^{6 \times n}$  in the case of  $\boldsymbol{\xi}$  and analytical Jacobian  $\mathbf{J}_a \in \mathbb{R}^{6 \times n}$  in the case of  $\mathbf{X}$ . The same approach is used to determine the transformation between the joint velocities and the linear and angular velocity of any point on the manipulator.

The two types of Jacobians, the geometric Jacobian  $\mathbf{J}_g$  and the analytical Jacobian  $\mathbf{J}_a$ , are derived differently and are used for different purposes. The geometric Jacobian  $\mathbf{J}_g$  is computed according to the geometric technique in which the contributions of each joint velocity are mapped to the end effector linear and angular velocity. But if the end effector pose is specified in terms of a minimal number of parameters in the operation space, such as the Euler angles  $\boldsymbol{\phi} = [\varphi, \vartheta, \psi]^T$  or the axis/angle representation  $(\mathbf{l}, \lambda)$ , a direct computation of the Jacobian via differentiation of the direct kinematics function with respect to the joint variables is preferred, which results in the analytical Jacobian  $\mathbf{J}_a$ . Where the geometric Jacobian  $\mathbf{J}_g$  is used to derive the manipulator dynamics and to relate end effector forces  $\mathbf{F} \in \mathbb{R}^6$  with joint torques  $\boldsymbol{\tau} \in \mathbb{R}^6$ , the analytical Jacobian  $\mathbf{J}_a$  is often used to describe end effector velocities  $\dot{\mathbf{X}}$  and accelerations  $\ddot{\mathbf{X}}$ .

### Manipulator Jacobian

For a given  $n$ -link manipulator the transformation from the base frame to the end effector frame can be described by

$$\mathbf{T}_n^0(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_n^0(\mathbf{q}) & \mathbf{o}_n^0(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.20)$$

where  $\mathbf{q} = [q_1, \dots, q_n]^T$  is the vector of joint variables. By using the concept of skew symmetric matrices as it is described in Appendix A.2 it is shown in [26] that the angular velocity vector  $\boldsymbol{\omega}_n^0$  of the end effector is defined by

$$\mathbf{S}(\boldsymbol{\omega}_n^0) = \dot{\mathbf{R}}_n^0(\mathbf{R}_n^0)^T \quad (2.21)$$

The linear velocity of the end effector is simply

$$\mathbf{v}_n^0 = \dot{\mathbf{o}}_n^0 \quad (2.22)$$

The manipulator Jacobian  $\mathbf{J} \in \mathbb{R}^{6 \times n}$  is then defined by the linear and angular velocity of the end effector as

$$\mathbf{v}_n^0 = \mathbf{J}_v \dot{\mathbf{q}} \quad (2.23)$$

$$\boldsymbol{\omega}_n^0 = \mathbf{J}_\omega \dot{\mathbf{q}} \quad (2.24)$$

with  $\mathbf{J}_v \in \mathbb{R}^{3 \times n}$  and  $\mathbf{J}_\omega \in \mathbb{R}^{3 \times n}$ . Equations (2.23) and (2.24) are written together as

$$\boldsymbol{\xi} = \mathbf{J} \dot{\mathbf{q}} \quad (2.25)$$

where

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{v}_n^0 \\ \boldsymbol{\omega}_n^0 \end{bmatrix} \quad \text{and} \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_\omega \end{bmatrix} \quad (2.26)$$

It is important to note that the velocity vector  $\boldsymbol{\xi}$  is not the derivative of a position variable, since the angular velocity vector  $\boldsymbol{\omega}_n^0$  is not the derivative of any particular time varying quantity. The following sections will describe the calculation of the geometric and analytical Jacobian.

### Geometric Jacobian

The geometric Jacobian  $\mathbf{J}_g(\mathbf{q})$ , is computed by using the homogeneous transformations  $\mathbf{A}_1, \dots, \mathbf{A}_n$ . It is therefore a natural choice after having derived the forward kinematics based on the DH convention. For the case of a  $n$ -link manipulator the angular part of the Jacobian,  $\mathbf{J}_{g,\omega}$ , is derived as

$$\mathbf{J}_{g,\omega} = [\mathbf{z}_0^0, \dots, \mathbf{z}_{n-1}^0] \quad (2.27)$$

where

$$\mathbf{z}_{i-1}^0 = \mathbf{R}_{i-1}^0 \mathbf{k} \quad (2.28)$$

with  $\mathbf{k} = [0, 0, 1]^T$ . The linear part of the Jacobian,  $\mathbf{J}_{g,v}$ , is given by

$$\mathbf{J}_{g,v_i} = \frac{\partial \mathbf{o}_n^0}{\partial q_i} = \mathbf{z}_{i-1}^0 \times (\mathbf{o}_n^0 - \mathbf{o}_{i-1}^0) \quad (2.29)$$

The geometric Jacobian  $\mathbf{J}_g(\mathbf{q})$  is then obtained by inserting Equations (2.27) and (2.29) into Equation (2.26).

### Analytical Jacobian

The analytical Jacobian  $\mathbf{J}_a(\mathbf{q})$  is based on a minimal representation for the orientation of the end effector frame, like the Euler angles or axis/angle representation. It is common to describe the end effector position and orientation by using minimal representations, e.g., for task space control. The analytical Jacobian is therefore used for determining the end effector velocity and acceleration of the end effector. The end effector pose is denoted by

$$\mathbf{X} = \begin{bmatrix} \mathbf{o}_n^0(\mathbf{q}) \\ \beta_n^0(\mathbf{q}) \end{bmatrix} \quad (2.30)$$

where  $\mathbf{o}_n^0(\mathbf{q})$  is the usual vector from the origin of the base frame to the origin of the end effector frame and  $\beta_n^0(\mathbf{q})$  denotes a minimal representation for the orientation of the end effector frame relative to the base frame. In the case of Euler angles the orientation vector is  $\beta = \phi = [\varphi, \vartheta, \psi]^T$ . To distinguish from the general manipulator Jacobian notion in Equation (2.25) this different notion is used to define the analytical Jacobian of the form

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{v}_n^0 \\ \dot{\beta}_n^0 \end{bmatrix} = \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}} \quad (2.31)$$

Given the Euler angle transformation  $\mathbf{R}_{ZYZ}$  and the skew symmetric matrix  $\mathbf{S}(\boldsymbol{\omega})$  it holds that

$$\dot{\mathbf{R}}_{ZYZ} = \mathbf{S}(\boldsymbol{\omega})\mathbf{R}_{ZYZ} \quad (2.32)$$

which results in an angular velocity  $\boldsymbol{\omega}$  as given by

$$\begin{aligned} \boldsymbol{\omega} &= \begin{bmatrix} c_\psi s_\vartheta \dot{\phi} - s_\psi \dot{\vartheta} \\ s_\psi s_\vartheta \dot{\phi} + c_\psi \dot{\vartheta} \\ \dot{\psi} + c_\vartheta \dot{\phi} \end{bmatrix} \\ &= \begin{bmatrix} c_\psi s_\vartheta & -s_\psi & 0 \\ s_\psi s_\vartheta & c_\psi & 0 \\ c_\vartheta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} = \mathbf{B}(\boldsymbol{\phi})\dot{\boldsymbol{\phi}} \end{aligned} \quad (2.33)$$

Using this transformation, the analytical Jacobian  $\mathbf{J}_a(\mathbf{q})$  can be computed from the geometric Jacobian  $\mathbf{J}_g(\mathbf{q})$  as

$$\mathbf{J}_a(\mathbf{q}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{-1}(\boldsymbol{\phi}) \end{bmatrix} \mathbf{J}_g(\mathbf{q}) \quad (2.34)$$

provided  $\det \mathbf{B}(\boldsymbol{\phi}) \neq 0$ .

The acceleration of the end effector relative to the base frame is obtained by differentiating Equation (2.31) which yields

$$\ddot{\mathbf{X}} = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \left( \frac{d}{dt} \mathbf{J}_a(\mathbf{q}) \right) \dot{\mathbf{q}} \quad (2.35)$$

For 6-DOF manipulators the inverse velocity and acceleration equations are given by

$$\dot{\mathbf{q}} = \mathbf{J}_a(\mathbf{q})^{-1} \dot{\mathbf{X}} \quad (2.36)$$

$$\ddot{\mathbf{q}} = \mathbf{J}_a(\mathbf{q})^{-1} \left[ \ddot{\mathbf{X}} - \left( \frac{d}{dt} \mathbf{J}_a(\mathbf{q}) \right) \dot{\mathbf{q}} \right] \quad (2.37)$$

provided  $\det \mathbf{J}_a(\mathbf{q}) \neq 0$ .

### Jacobian for an Arbitrary Point on a Link

For the derivation of the manipulator dynamics it is necessary to derive the Jacobian not only for the end effector, but also for arbitrary points on a link.

Given a vector  $\mathbf{r}_{Pj}^0$  that is pointing to a point  $P$  on link  $l_j$ , expressed in the inertial coordinate frame  $o_0x_0y_0z_0$ , the elements of the linear part of the Jacobian  $\mathbf{J}_{Pj}(\mathbf{q})$  are

$$\begin{aligned}\mathbf{J}_{Pj,\mathbf{v}} &= \frac{\partial \mathbf{r}_{Pj}^0}{\partial \mathbf{q}_i} = \mathbf{z}_{i-1}^0 \times (\mathbf{r}_{Pj}^0 - \mathbf{o}_{i-1}^0), \quad \forall i \leq j \\ \mathbf{J}_{Pj,\mathbf{v}} &= \mathbf{0}, \quad \forall i > j\end{aligned}\quad (2.38)$$

for  $i \in \{1, \dots, n\}$ . The columns of  $\mathbf{J}_{Pj,\mathbf{v}}$  for  $i > j$  are zero since the velocity of the  $j^{\text{th}}$  link is unaffected by the motion of the links that come after it. The same holds for the angular part of the Jacobian

$$\begin{aligned}\mathbf{J}_{Pj,\boldsymbol{\omega}} &= \mathbf{z}_{i-1}^0, \quad \forall i \leq j \\ \mathbf{J}_{Pj,\boldsymbol{\omega}} &= \mathbf{0}, \quad \forall i > j\end{aligned}\quad (2.39)$$

The Jacobian  $\mathbf{J}_{Pj}(\mathbf{q})$  is then

$$\mathbf{J}_{Pj}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0^0 \times (\mathbf{r}_{Pj}^0 - \mathbf{o}_0^0) & \dots & \mathbf{z}_{j-1}^0 \times (\mathbf{r}_{Pj}^0 - \mathbf{o}_{j-1}^0) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{z}_0^0 & \dots & \mathbf{z}_{j-1}^0 & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \quad (2.40)$$

Note that the vector  $\mathbf{r}_{Pj}^0$  must be computed for each link as it is not given directly by the homogeneous transformation matrices  $\mathbf{A}_1, \dots, \mathbf{A}_n$  or  $\mathbf{T}$ .

### Velocity Transformation

The transformation of velocities between two coordinate frames is derived in [24]. The transformation of velocities between a frame  $o_i x_i y_i z_i$  and a frame  $o_j x_j y_j z_j$  can be described by

$$\begin{bmatrix} \dot{\mathbf{p}}_j^j \\ \boldsymbol{\omega}_j^j \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^j & -\mathbf{R}_i^j \mathbf{S}(\mathbf{r}_{ij}^i) \\ \mathbf{0} & \mathbf{R}_i^j \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_i^i \\ \boldsymbol{\omega}_i^i \end{bmatrix} \quad (2.41)$$

where  $\mathbf{r}_{ij}^i$  is a vector pointing from frame  $o_i x_i y_i z_i$  to frame  $o_j x_j y_j z_j$ , expressed in frame  $o_i x_i y_i z_i$ , and  $\mathbf{S}(\cdot)$  is a skew-symmetric matrix as defined in Appendix A.2.

The following section presents methods that can be used to approximate inertia tensors for manipulator links.

### 2.2.4. Inertia Tensors

To derive the dynamic equations of a manipulator it is necessary to know the inertia tensor of each link. They are often not provided by the manufacturer and have to be calculated.

The general inertia tensor has the form

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (2.42)$$

where  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are the principal moments of inertia and where  $I_{xy} = I_{yx}$ ,  $I_{xz} = I_{zx}$  and  $I_{yz} = I_{zy}$  are the products of inertia. They are calculated by

$$\begin{aligned} I_{xx} &= \iiint (y^2 + z^2) \rho(x, y, z) \, dx \, dy \, dz \\ I_{yy} &= \iiint (x^2 + z^2) \rho(x, y, z) \, dx \, dy \, dz \\ I_{zz} &= \iiint (x^2 + y^2) \rho(x, y, z) \, dx \, dy \, dz \end{aligned} \quad (2.43)$$

and

$$\begin{aligned} I_{xy} = I_{yx} &= - \iiint xy \rho(x, y, z) \, dx \, dy \, dz \\ I_{xz} = I_{zx} &= - \iiint xz \rho(x, y, z) \, dx \, dy \, dz \\ I_{yz} = I_{zy} &= - \iiint yz \rho(x, y, z) \, dx \, dy \, dz \end{aligned} \quad (2.44)$$

where  $\rho(x, y, z)$  is the mass density of the object, represented as a function of position.

The centroid attached frame  $o_C x_C y_C z_C$  is a frame that is located at the centroid  $C$  of a body. For the case where the inertia tensor has to be calculated around a coordinate frame that is parallel to the centroid frame, but not aligned with it, the parallel axis theorem can be used. The inertia tensor of the cylinder with respect to the centroid attached frame is denoted by  $\mathbf{I}_C$  and the parallel frame is  $oxyz$ . A vector  $\mathbf{p}_C$  is pointing to  $C$ , expressed in frame  $oxyz$ . The parallel frame is then shifted by a distance  $\mathbf{o}_{p_C} = [o_{p_C,x}, o_{p_C,y}, o_{p_C,z}] = \mathbf{p}_C$

relative to the centroid frame, expressed in the frame  $oxyz$ . The principal moments of inertia for a body with mass  $m$  and around frame  $oxyz$  can then be calculated by

$$\begin{aligned} I_{xx} &= I_{C,xx} + m \left( o_{\mathbf{p}_C,y}^2 + o_{\mathbf{p}_C,z}^2 \right) \\ I_{yy} &= I_{C,yy} + m \left( o_{\mathbf{p}_C,x}^2 + o_{\mathbf{p}_C,z}^2 \right) \\ I_{zz} &= I_{C,zz} + m \left( o_{\mathbf{p}_C,x}^2 + o_{\mathbf{p}_C,y}^2 \right) \end{aligned} \quad (2.45)$$

and the products of inertia by

$$\begin{aligned} I_{xy} = I_{yx} &= I_{C,xy} - m o_{\mathbf{p}_C,x} o_{\mathbf{p}_C,y} \\ I_{xz} = I_{zx} &= I_{C,xz} - m o_{\mathbf{p}_C,x} o_{\mathbf{p}_C,z} \\ I_{yz} = I_{zy} &= I_{C,yz} - m o_{\mathbf{p}_C,y} o_{\mathbf{p}_C,z} \end{aligned} \quad (2.46)$$

### Inertia Tensor for Cylinders with Constant Mass Density

In the following it is assumed that the mass  $M$  of each link is known and that a mass center point  $p_M$ , pointed to by  $\mathbf{p}_M = [x_M, y_M, z_M]^T$ , is provided for each link. The links can be furthermore approximated by cylinders with a radius  $r$  and a height  $h$ . The mass center point  $p_M$  is at the geometric center  $C$  of the cylinder, which is pointed to by  $\mathbf{p}_C = [x_C, y_C, z_C]^T$ . The mass density is assumed to be constant, i.e.,  $\rho(x, y, z) = \rho = \frac{M}{\pi r^2 h} = \text{const.}$  and the resulting inertia tensor calculated around the body attached frame is then

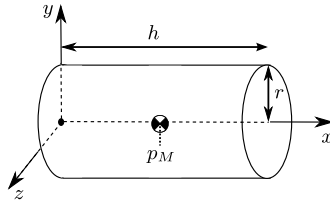
$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.47)$$

If the axis of symmetry of the cylinder is aligned with the  $x$ -axis, as visualized in Figure 2.3, the moments of inertia are

$$\begin{aligned} I_{xx} &= \frac{1}{2} M r^2 \\ I_{yy} = I_{zz} &= \frac{1}{12} M (3r^2 + h^2) \end{aligned} \quad (2.48)$$

An inertia tensor with these moments of inertia is denoted by  $\mathbf{I}_{\text{cyl},x}$  for the remaining part of this thesis. For cylinders that have a symmetric axis aligned to the  $y$ - or  $z$ -axis, the inertia tensors are then  $\mathbf{I}_{\text{cyl},y}$  and  $\mathbf{I}_{\text{cyl},z}$ , respectively.





**Figure 2.3.:** Cylinder with height  $h$  and radius  $r$ . Its mass center point  $p_M$  is at the geometric center  $C$ . The mass density  $\rho(x, y, z)$  of the cylinder is assumed to be evenly distributed and therefore constant, i.e.,  $\rho = \text{const}$ .

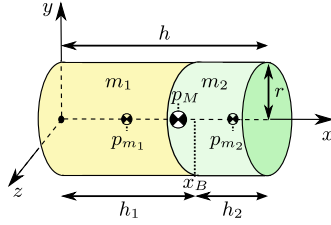
### Inertia Tensor for Cylinders with Varying Mass Density

A more complicated case arises when the mass center point  $p_M$  is not at the same position as the geometric center  $C$ . Especially when the geometric shape of the link can still be approximated by a symmetric body, a shifted mass center point  $p_M$  indicates that the mass density  $\rho(x, y, z)$  cannot be assumed to be constant throughout the link. As the mass density of the links are unknown and therefore the calculation of the inertia tensor not possible, further approximations must be made.

Nevertheless, it is common to model the links as cylinders where the mass center point  $p_M$  is aligned with the geometric center  $C$  [15, 21]. This carries the risk that the dynamic model becomes far less accurate. Another attempt to obtain more accurate inertia tensors is derived in this thesis. It keeps the simplicity of using basic geometric objects to approximate the shape of the link and the assumption that each object has a constant density. But instead of using only one object for each link, several geometric shapes are combined to resemble the link shape while the combined mass center point is at the same position as provided by the manufacturer. This method is based on heuristically obtained knowledge about the mass distribution in the links. But in the case that it is applied to a manipulator with unknown inertia tensors, well approximated inertia tensors can be calculated by using the parallel axis theorem. The method is presented in the following section.

### Reconstruct Mass Center Point with Multiple Cylinders

A link with a mass  $M$  is given, which can be approximated by a cylinder with height  $h$  and radius  $r$ . It has a mass center point  $p_M$ , pointed to by



**Figure 2.4.:** Cylinder with height  $h$  and radius  $r$ . Its mass center point  $p_M$  is not at the geometric center  $C$ . A reconstruction of the mass center point  $p_M$  is achieved by splitting the cylinder into two cylinders with homogeneous mass distribution. The masses  $m_1$ ,  $m_2$  and heights  $h_1$ ,  $h_2$  are chosen such that the mass center points  $p_{m_1}$  and  $p_{m_2}$  combined resemble the initial mass center point  $p_M$ .

$\mathbf{p}_M = [x_M, y_M, z_M]^T$ , which is not aligned with the centroid  $C$ , as visualized in Figure 2.4. The mass center point lies on the axis of symmetry, such that  $y_M = y_C$  and  $z_M = z_C$ , but  $x_M \neq x_C$ . In this specific case the cylinder can be split in two cylinders with heights  $h_1$  and  $h_2$ , radii  $r_1$  and  $r_2$  and masses  $m_1$  and  $m_2$ , respectively, where

$$M = m_1 + m_2 \quad (2.49)$$

Note that the radii are  $r_1 = r_2 = r$  in this case. The two cylinders have a mass center point  $p_{m_1}$  and  $p_{m_2}$ , which are pointed to by  $\mathbf{p}_{m_1} = [x_{m_1}, y_{m_1}, z_{m_1}]^T$  and  $\mathbf{p}_{m_2} = [x_{m_2}, y_{m_2}, z_{m_2}]^T$ , respectively, and where  $y_{m_1} = y_{m_2} = y_M = y_C$  and  $z_{m_1} = z_{m_2} = z_M = z_C$ . The relationship between  $\mathbf{p}_M$ ,  $\mathbf{p}_{m_1}$  and  $\mathbf{p}_{m_2}$  can be described by

$$\mathbf{p}_M = \frac{\mathbf{p}_{m_1} m_1 + \mathbf{p}_{m_2} m_2}{m_1 + m_2} \quad (2.50)$$

where in this special case

$$x_M = \frac{x_{m_1} m_1 + x_{m_2} m_2}{m_1 + m_2} \quad (2.51a)$$

$$y_M = y_{m_1} = y_{m_2} \quad (2.51b)$$

$$z_M = z_{m_1} = z_{m_2} \quad (2.51c)$$

Assuming that there is a border between the two cylinders which can be defined by a plane  $\Pi_B$  normal to the  $x$ -axis, crossing it at  $x_B$ , the position of

$x_{m_1}$  and  $x_{m_2}$  can be written as

$$x_{m_1} = \frac{x_B}{2} \quad (2.52a)$$

$$x_{m_2} = x_B + \frac{h - x_B}{2} = \frac{1}{2}(h + x_B) \quad (2.52b)$$

Substituting Equations (2.52) into Equation (2.51a) results in

$$\begin{aligned} x_M &= \frac{\frac{x_B}{2}m_1 + \frac{1}{2}(h + x_B)m_2}{m_1 + m_2} \\ &= \frac{1}{2} \frac{x_B(m_1 + m_2) + hm_2}{m_1 + m_2} \\ &= \frac{1}{2} \left( x_B + \frac{hm_2}{m_1 + m_2} \right) \end{aligned} \quad (2.53)$$

As there are three unknown variables, i.e.,  $m_1$ ,  $m_2$ , and  $x_B$ , and only two independent equations, i.e., Equations (2.49) and (2.53), it is necessary to make further assumptions. In this thesis the masses  $m_1$  and  $m_2$  are chosen heuristically based on knowledge about the probable mass distribution in a link. Now that  $m_1$  and  $m_2$  are chosen,  $x_B$  can be calculated by

$$x_B = 2x_M - \frac{hm_2}{M} \quad (2.54)$$

and the heights of the cylinders are

$$h_1 = x_B \quad (2.55a)$$

$$h_2 = h - x_B \quad (2.55b)$$

This results in an estimation where two cylinders combined together resemble the shape of the link and reconstruct a mass center point as provided by the manufacturer. An inertia tensor of a combined body is equal to the sum of the inertia tensors of the bodies it is made from, as long as all inertia tensors are calculated around a common coordinate frame. The inertia tensor of the approximated link, with respect to the body attached frame  $o_M x_M y_M z_M$  and denoted by  $\mathbf{I}_M$ , can therefore be calculated by

$$\mathbf{I}_M = \mathbf{I}_{M,1} + \mathbf{I}_{M,2} \quad (2.56)$$

where the inertia tensors  $\mathbf{I}_{M,1}$  and  $\mathbf{I}_{M,2}$  are the inertia tensors of the two divided cylinders, likewise calculated around the body attached frame  $o_M x_M y_M z_M$ .

Those inertia tensors are calculated by first calculating the inertia tensors of the divided cylinders in their local body attached frames according to Equations (2.47) and (2.48), such that

$$\mathbf{I}_{m_1} = \mathbf{I}_{\text{cy1},x} \quad (2.57a)$$

$$\mathbf{I}_{m_2} = \mathbf{I}_{\text{cy1}_2,x} \quad (2.57b)$$

Then as the distances  $\mathbf{o}_{p_{m_1}}^{p_M}$  and  $\mathbf{o}_{p_{m_2}}^{p_M}$  are known to be

$$\mathbf{o}_{p_{m_1}}^{p_M} = -\mathbf{p}_M + \mathbf{p}_{m_1} \quad (2.58a)$$

$$\mathbf{o}_{p_{m_2}}^{p_M} = -\mathbf{p}_M + \mathbf{p}_{m_2} \quad (2.58b)$$

the inertia tensors  $\mathbf{I}_{m_1}$  and  $\mathbf{I}_{m_2}$  can be calculated in frame  $o_M x_M y_M z_M$  by using the parallel axis theorem as given in Equations (2.45) and (2.46). They are then denoted as  $\mathbf{I}_{M,1}$  and  $\mathbf{I}_{M,2}$ .

This method is easily extended for more complicated link shapes and the use of more than two bodies to approximate the links and their mass distributions.

The following section presents the derivation of dynamic equations for industrial manipulators by using forward and velocity kinematics and inertia tensors.

### 2.2.5. Dynamics

The dynamics of a robot describe the relationship between forces, torques and motion. Two methods are commonly used for the derivation of the dynamic equations: the Euler-Lagrange equations and the Newton-Euler formulation. The Euler-Lagrange method is derived from D'Alembert's principle and the principle of virtual work, whereas the Newton-Euler formulation is a recursive formulation of the dynamic equations that is often used for numerical calculation.

The Euler-Lagrange method was chosen in this work. Whilst the derivation of the dynamic equations for an  $n$ -link manipulator by means of the Euler-Lagrange equations can be found in [26], only the substantial kinetic and potential energy equations and eventually the dynamic equations are presented in the following section.

#### Kinetic Energy for an $n$ -Link Manipulator

Given a rigid body with a body attached coordinate frame, the kinetic energy of that body is the sum of the translational and rotational kinetic energy. The

translational energy is obtained by concentrating the entire mass of the object at the center of mass, while the rotational energy is calculated about the center of mass. The total kinetic energy is then

$$\mathcal{K} = \frac{1}{2}m\mathbf{v}^T\mathbf{v} + \frac{1}{2}\boldsymbol{\omega}^T\mathcal{I}\boldsymbol{\omega} \quad (2.59)$$

where  $m \in \mathbb{R}$  is the total mass of the object,  $\mathbf{v} \in \mathbb{R}^3$  and  $\boldsymbol{\omega} \in \mathbb{R}^3$  are the linear and angular velocity vectors, respectively, and  $\mathcal{I} \in \mathbb{R}^{3 \times 3}$  is the inertia tensor expressed in the inertial frame. The inertia tensor  $\mathcal{I}$  can be written as

$$\mathcal{I} = \mathbf{R}\mathbf{I}\mathbf{R}^T \quad (2.60)$$

where  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  is the rotation matrix from the inertial frame to the body attached frame, and where  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  is the inertia tensor expressed in the body attached frame. Given the assumption that each link  $l_i$ ,  $i \in \{1, \dots, n\}$ , is a rigid body, and by using the Equations (2.23), (2.24) and (2.60), the overall kinetic energy of the manipulator equals

$$K = \frac{1}{2}\dot{\mathbf{q}}^T \underbrace{\left[ \sum_{i=1}^n \{m_i \mathbf{J}_{\mathbf{v}_{mi}}(\mathbf{q})^T \mathbf{J}_{\mathbf{v}_{mi}}(\mathbf{q}) + \mathbf{J}_{\boldsymbol{\omega}_{mi}}(\mathbf{q})^T \mathbf{R}_i(\mathbf{q}) \mathbf{I}_{mi} \mathbf{R}_i(\mathbf{q})^T \mathbf{J}_{\boldsymbol{\omega}_{mi}}(\mathbf{q})\} \right]}_{\mathbf{D}(\mathbf{q})} \dot{\mathbf{q}} \quad (2.61)$$

where  $m_i \in \mathbb{R}$  is the mass of link  $l_i$  and where  $\mathbf{J}_{\mathbf{v}_{mi}} \in \mathbb{R}^{3 \times n}$  and  $\mathbf{J}_{\boldsymbol{\omega}_{mi}} \in \mathbb{R}^{3 \times n}$  are the linear and angular parts of the Jacobian matrix from base to the mass center of link  $l_i$  as described in Equation (2.40).  $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$  is the rotation matrix from base to link  $l_i$  and  $\mathbf{I}_{mi} \in \mathbb{R}^{3 \times 3}$  is the inertia tensor expressed in the body attached frame of link  $l_i$ .  $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is the symmetric and positive definite inertia matrix of the manipulator.

### Potential Energy for an n-Link Manipulator

In the case of rigid dynamics, the only source of potential energy is gravity. The potential energy of the  $i^{th}$  link can be computed by assuming that the mass of the entire object is concentrated at its center of mass and is given by

$$P_i = m_i \mathbf{g}_0^T \mathbf{r}_{mi} \quad (2.62)$$

where  $m_i$  is the mass of link  $l_i$ ,  $\mathbf{g}_0$  is the vector giving the direction of gravity in the inertial frame and the vector  $\mathbf{r}_{mi}$  gives the coordinates of the center of mass of link  $i$ . The total potential energy of the  $n$ -link robot is therefore

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i \mathbf{g}_0^T \mathbf{r}_{mi} \quad (2.63)$$

### Equations of Motion

It is common to write the equations of motion in the form as

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.64)$$

where  $\mathbf{D} \in \mathbb{R}^{n \times n}$  is the inertia matrix,  $\mathbf{C} \in \mathbb{R}^{n \times n}$  is the coriolic and centrifugal matrix,  $\mathbf{g} \in \mathbb{R}^n$  is the gravity vector,  $\boldsymbol{\tau} \in \mathbb{R}^n$  is the torque vector and  $\mathbf{q} \in \mathbb{R}^n$  and  $\dot{\mathbf{q}} \in \mathbb{R}^n$  are the position and velocity vectors, respectively.

The inertia matrix  $\mathbf{D}(\mathbf{q})$  is defined in Equation (2.61). The  $(k, j)^{th}$  element of the matrix  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is defined as

$$c_{kj} = \sum_{i=1}^n c_{ijk}(\mathbf{q})\dot{q}_i \quad (2.65)$$

where  $c_{ijk}$  are the Christoffel symbols

$$c_{ijk} := \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \quad (2.66)$$

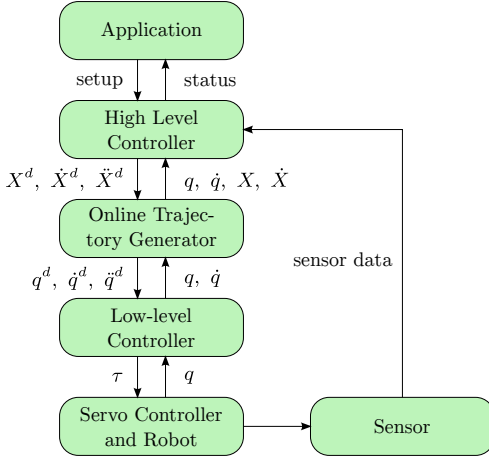
and where  $d_{ij}$  are the entries of the inertia matrix. Finally, the gravity vector entries are defined by

$$g_k = \frac{\partial P}{\partial q_k} \quad (2.67)$$

The control of industrial manipulators and the problematic of a closed control architecture are discussed in the following section.

## 2.3. Control of Industrial Manipulators

The control architecture for industrial manipulators often follows a hierarchical structure as it is visualized in Figure 2.5. How many of the layers and how much sensor data are accessible to the user depends on the manufacturer and therefore varies greatly between the industrial manipulators. The lowest layer,



**Figure 2.5.:** Hierarchical control structure as it is often used for industrial manipulators [23]. The accessibility to the separate control layers and the available sensors differ greatly between manufacturers.

namely the server controller and robot, refers to the independent joint control as well as the multivariable control of the manipulator. The low-level controller processes trajectories in joint space to the necessary torque in the joints. On top of that controller the trajectories in joint space are generated online from the given trajectories in task space. Upmost is the application layer which transforms different tasks to motion in task space control.

Independent joint control is used to control the torque output based on the voltage input. It uses a model of the motor in the joints and does not take the manipulator dynamics into account. A commonly chosen control method for independent joint control is the PID controller.

Multivariable control controls either the joint position of all joints  $q_i$ ,  $i \in \{1, \dots, n\}$ , or the end effector position and orientation  $\mathbf{X}(q)$ , based on the torque  $\tau$  as an input. Those two areas are called the joint space and task space control, respectively. In both cases the dynamic model of the manipulator is taken into account and inverse dynamics are often used to address the nonlinearity. A PD controller is then used to control the linearized system.

To approach the problem where contact forces are present force control is

used. Force Control allows a flexible and sensible interaction with the environment in both manufacturing applications and the human-machine interactions.

The control method relevant to this thesis is multivariable control and will be introduced in the following section.

### 2.3.1. Multivariable Control

A special form of multivariable control is considered in this thesis, namely trajectory tracking by using feedback linearization. It is furthermore divided into joint space control and task space control and uses the method of inverse dynamics.

#### Joint Space Inverse Dynamics

Considering the dynamic equations of an  $n$ -link manipulator as given in Equation (2.64) and choosing the torque as the control input  $\mathbf{u} = \boldsymbol{\tau} \in \mathbb{R}^n$ , then a nonlinear feedback control law

$$\mathbf{u} = f(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (2.68)$$

is sought such that the overall closed-loop system is linear. In the case of the manipulator dynamics given by Equation (2.64) the problem is easy and the solution is to choose the control vector  $\mathbf{u}$  according to the equation

$$\mathbf{u} = \mathbf{D}(\mathbf{q})\mathbf{a}_q + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (2.69)$$

Since the inertia matrix  $\mathbf{D}$  is invertible, the combined system given by Equations (2.64), (2.68) and (2.69) reduces to

$$\ddot{\tilde{\mathbf{q}}} = \mathbf{a}_q \quad (2.70)$$

which represents a double integrator system and where  $\mathbf{a}_q$  is the new input that is yet to be chosen. A choice for  $\mathbf{a}_q$  is a PD controller with feedforward acceleration as

$$\mathbf{a}_q = \ddot{\tilde{\mathbf{q}}}^d(t) - \mathbf{K}_0\tilde{\mathbf{q}} - \mathbf{K}_1\dot{\tilde{\mathbf{q}}} \quad (2.71)$$

where  $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}^d$ ,  $\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}^d$  and where  $\mathbf{K}_0$ ,  $\mathbf{K}_1$  are diagonal matrices with diagonal elements consisting of position and velocity gains, respectively. The reference trajectory

$$t \rightarrow (\mathbf{q}^d(t), \dot{\mathbf{q}}^d(t), \ddot{\mathbf{q}}^d(t)) \quad (2.72)$$



defines the desired time history of joint positions, velocities, and accelerations. The gain matrices  $\mathbf{K}_0$  and  $\mathbf{K}_1$  can be chosen as

$$\mathbf{K}_0 = \begin{bmatrix} \kappa_{0,1} & 0 & \dots & 0 \\ 0 & \kappa_{0,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \kappa_{0,n} \end{bmatrix}, \quad \mathbf{K}_1 = \begin{bmatrix} \kappa_{1,1} & 0 & \dots & 0 \\ 0 & \kappa_{1,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \kappa_{1,n} \end{bmatrix} \quad (2.73)$$

which results in a decoupled closed-loop system. In order to stabilize the system, the control parameters have to be positive. By choosing

$$\begin{aligned} \kappa_{0,i} &= \omega_{n,i}^2 \\ \kappa_{1,i} &= 2\omega_{n,i} \end{aligned} \quad \forall i = 1, \dots, n \quad (2.74)$$

each joint response will be equal to the response of a critically damped linear second order system with natural frequency  $\omega_{n,i}$ . Figure 2.6a visualizes the closed loop joint space control.

### Task Space Inverse Dynamics

Given the end effector pose  $\mathbf{X} \in \mathbb{R}^6$  as defined in Equation (2.30) and the end effector velocity  $\dot{\mathbf{X}} \in \mathbb{R}^6$  as defined in Equation (2.31) the acceleration at the end effector is

$$\ddot{\mathbf{X}} = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}} \quad (2.75)$$

By rewriting Equation (2.75) as

$$\ddot{\mathbf{q}} = \mathbf{J}_a^{-1} [\ddot{\mathbf{X}} - \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}] \quad (2.76)$$

and considering the double integrator system of Equation (2.70) a good choice is to set

$$\mathbf{a}_q = \mathbf{J}_a^{-1} [\mathbf{a}_X - \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}] \quad (2.77)$$

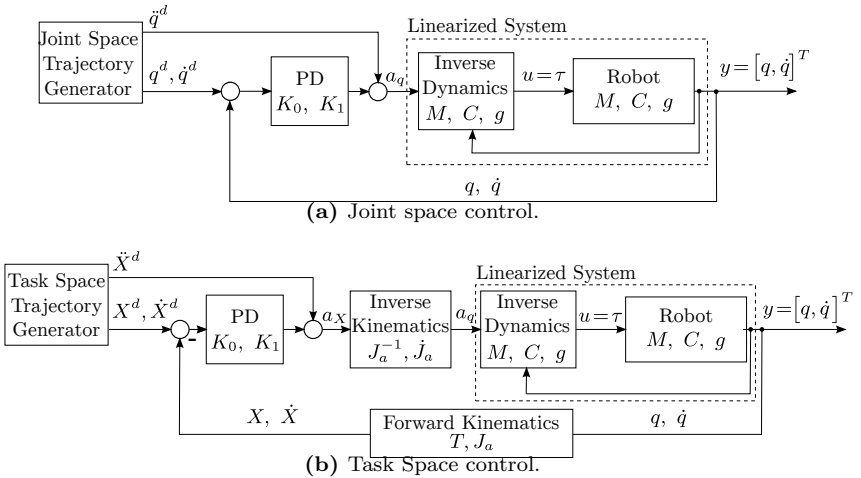
and to obtain a double integrator system in task space coordinates

$$\ddot{\mathbf{X}} = \mathbf{a}_X \quad (2.78)$$

A choice for  $\mathbf{a}_X$  is

$$\mathbf{a}_X = \ddot{\mathbf{X}}^d(t) - \mathbf{K}_0\tilde{\mathbf{X}} - \mathbf{K}_1\dot{\tilde{\mathbf{X}}} \quad (2.79)$$

where  $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{X}^d$ ,  $\dot{\tilde{\mathbf{X}}} = \dot{\mathbf{X}} - \dot{\mathbf{X}}^d$  and where  $\mathbf{K}_0$ ,  $\mathbf{K}_1$  are diagonal matrices with diagonal elements consisting of position and velocity gains, respectively. Figure 2.6b visualizes the closed loop task space control.

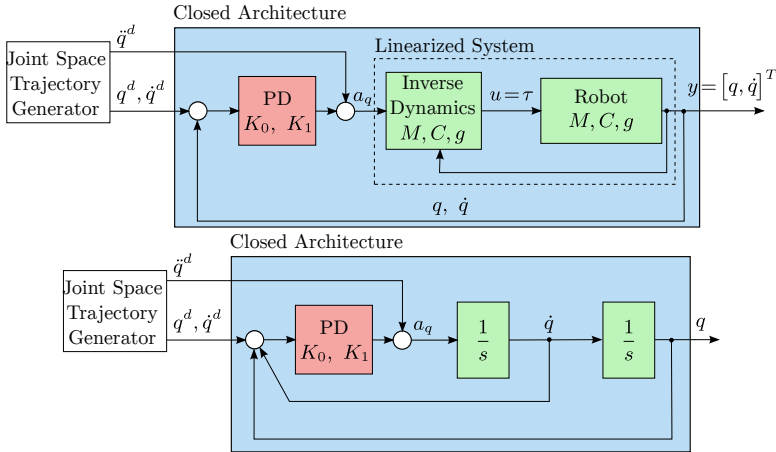


**Figure 2.6.:** Common joint and task space control schemes for industrial manipulators.

The problematic of a closed control architecture and the idea to estimate PD control parameters in order to get a model of the closed loop are presented in the next section.

### 2.3.2. Closed Control Architecture

In the case that access to one or several control layers of the control structure of Figure 2.5 is not given by the manufacturer, it might not be possible to get an insight into the implemented controllers and their parameters. In that case it will not be possible to verify the model in an open loop. Instead the model has to be verified in a closed loop where assumptions have to be made about the implemented controllers. The success of the model verification is based on the available measurements and correctly guessing the underlying controllers. A parameter estimation of control parameters will at least ensure that the closed loop model behaves in the same way as the manipulator closed loop system.



**Figure 2.7.:** Common joint space control scheme for industrial manipulators, subject to a closed control architecture. The user can merely input position, velocity and acceleration trajectories and receives position and velocity feedback. The dynamic equations are modeled, whereas the PD control parameters are unknown. The linearized system reduces to a double integrator system.

### PD Parameter Estimation

A first guess about the implemented joint space and task space controllers is that the most common controllers as presented in Section 2.3 are used. Assuming that the PD parameters are the only unknown parameters in the closed loop they can be estimated by using the method of Nonlinear Least Squares (NLS).

### Nonlinear Least Squares

Considering the equations of motion in a closed loop

$$\begin{aligned}
 D(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)) &= \tau(t) \\
 \tau(t) &= D(q(t))a_q(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)) \\
 a_q(t) &= \ddot{q}^d(t) - K_0(q - q^d) - K_1(\dot{q} - \dot{q}^d)
 \end{aligned} \tag{2.80}$$

with the unknown matrices  $\mathbf{K}_0$  and  $\mathbf{K}_1$ , the available measurements are

$$\mathbf{y}(t, \mathbf{K}_0, \mathbf{K}_1) = \begin{bmatrix} \mathbf{q}(t, \mathbf{K}_0, \mathbf{K}_1, \boldsymbol{\tau}(t)) \\ \dot{\mathbf{q}}(t, \mathbf{K}_0, \mathbf{K}_1, \boldsymbol{\tau}(t)) \end{bmatrix} \quad (2.81)$$

where  $\mathbf{D} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{C} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{g} \in \mathbb{R}^n$ ,  $\boldsymbol{\tau} \in \mathbb{R}^n$ ,  $\mathbf{q} \in \mathbb{R}^n$ ,  $\dot{\mathbf{q}} \in \mathbb{R}^n$ ,  $\mathbf{K}_0 \in \mathbb{R}^{n \times n}$ ,  $\mathbf{K}_1 \in \mathbb{R}^{n \times n}$  and  $\mathbf{y} \in \mathbb{R}^{2n}$ . As there are a total of  $2n$  parameters that define  $\mathbf{K}_0$  and  $\mathbf{K}_1$ , according to Equation (2.73), only  $\boldsymbol{\kappa} = [\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1] \in \mathbb{R}^{2n}$  needs to be estimated and Equation (2.81) can be rewritten to

$$\mathbf{y}(t, \boldsymbol{\kappa}) = \begin{bmatrix} \mathbf{q}(t, \boldsymbol{\kappa}, \boldsymbol{\tau}(t)) \\ \dot{\mathbf{q}}(t, \boldsymbol{\kappa}, \boldsymbol{\tau}(t)) \end{bmatrix} \quad (2.82)$$

In order to estimate the parameters  $\boldsymbol{\kappa}$ ,  $N$  measurements  $\mathbf{y}_m(t_j)$  are taken at the time instances  $t_j$ , for  $j = 1, \dots, N$  and an optimization problem is set up to

$$\min_{\boldsymbol{\kappa}} \mathcal{J}(\boldsymbol{\kappa}) = \sum_{i=1}^n \sum_{j=1}^N (y_i(t_j, \boldsymbol{\kappa}) - y_{m,i}(t_j))^2 \quad (2.83)$$

s.t.  $\mathbf{y}(t_j, \boldsymbol{\kappa})$  is the solution of system (2.80)

where  $n$  is the number of measurements available. The vector  $\boldsymbol{\kappa}$  that minimizes  $\mathcal{J}(\boldsymbol{\kappa})$  is the Nonlinear Least Squares Estimator (NLSE) and is written as

$$\hat{\boldsymbol{\kappa}} = \min_{\boldsymbol{\kappa}} \mathcal{J}(\boldsymbol{\kappa}) \quad (2.84)$$

## 2.4. Summary

The well defined rigid motions and DH parameters simplify the derivation of the forward and velocity kinematics. The derivation of the geometric Jacobian is necessary in order to obtain the velocity kinematics and dynamic equations, whereas the analytical Jacobian is used to calculate the velocity and acceleration of the end effector. The calculation of the inertia tensors can be difficult if little information about the manipulator links is given. If the mass center points of each link are known, it is possible to incorporate this information by modeling the links with multiple bodies with constant mass density instead of using single-body models. The equations of motion based on Euler-Lagrange equations are easily derived when forward and velocity kinematics and inertia tensors are known.

Multivariable control for industrial manipulators in both joint- and task-space is commonly based on feedback linearization and a PD controller. The implementation and PD parameters can be hidden from the user if the control architecture is restricted by the manufacturer. A closed loop model can be obtained by estimating the PD control parameters with a NLSE.



## 3. Principles of Force Estimation

This chapter introduces the main aspects of force estimation. In order to obtain correct force estimates, it is important to avoid singular configurations of the manipulator. Methods to find the singular configurations are therefore explained in Section 3.1. Section 3.2 introduces the basics of force estimation and a simple method to estimate forces/torques. Friction modeling is important for successful force estimation, which is why the Coulomb friction model and viscous friction model are presented in Section 3.2.1.

### 3.1. Singularities

In a configuration  $\mathbf{q}$  where the Jacobian  $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times n}$  loses rank, i.e.,  $\text{rank } \mathbf{J}(\mathbf{q}) < 6$ , the motion of the robot can be constrained in many ways. Such a configuration is called a singular configuration. Singularities represent configurations from which certain directions of motion may be unattainable, where bounded end effector velocities may correspond to unbounded joint velocities and where bounded joint torques may correspond to unbounded end effector forces and torques. The latter case does especially affect the force estimation. Singular configurations should be therefore avoided, if possible.

In the case that  $n = 6$ , the Jacobian is a square matrix and singular if and only if  $\det \mathbf{J}(\mathbf{q}) = 0$ . In general, it is difficult to solve this nonlinear equation and to obtain the singularities. Simplifications in the calculation are possible for manipulators with a spherical wrist. This is not the case in this study, such that the calculation of the singularities from the manipulator Jacobian is beyond its scope.

When working with the analytical Jacobian  $\mathbf{J}_a$ , additional singularities emerge. This is due to the relation of  $\mathbf{J}_a$  with the geometric Jacobian  $\mathbf{J}_g$ , see Equation (2.34), where  $\mathbf{B}^{-1}(\phi)$  is required. These additional singularities are called representational singularities and can cause problems, for example in the calculation of end effector accelerations, which requires  $\frac{d}{dt} \mathbf{J}_a$ .

The analytical Jacobian  $\mathbf{J}_a(\mathbf{q})$  can be computed from the geometric Jacobian  $\mathbf{J}_g(\mathbf{q})$  using Equation (2.34), provided  $\det \mathbf{B}(\phi) \neq 0$ . Representational

singularities arise when

$$\det \mathbf{B}(\phi) = \det \begin{bmatrix} c_\psi s_\vartheta & -s_\psi & 0 \\ s_\psi s_\vartheta & c_\psi & 0 \\ c_\vartheta & 0 & 1 \end{bmatrix} = c_\psi^2 s_\vartheta + s_\psi^2 s_\vartheta = s_\vartheta = 0 \quad (3.1)$$

which is the case for  $\vartheta \in \{k\pi\}$ ,  $k \in \mathbb{Z}$ . The singularities of the analytical Jacobian are the union of the manipulator Jacobian singularities and representational singularities.

The principles of force estimation, a simple force estimation method and friction modeling are presented in the next section.

## 3.2. Force Estimation

Interaction of the manipulator with the environment produces forces and moments at the end effector or tool. These, in turn, produce torques at the joints of the robot. The manipulator Jacobian puts a quantitative relationship between the end effector forces/torques and joint torques.

Let  $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z]^T$  represent the vector of forces and moments at the end effector, expressed in the base frame. Let  $\boldsymbol{\tau}_{ext}$  denote the corresponding vector of joint torques. Then  $\mathbf{F}$  and  $\boldsymbol{\tau}_{ext}$  are related by

$$\boldsymbol{\tau}_{ext} = \mathbf{J}_g^T(\mathbf{q})\mathbf{F} \quad (3.2)$$

In cases where it is possible to estimate the joint torques that are caused by external forces/torques, denoted by  $\hat{\boldsymbol{\tau}}_{ext}$ , the external forces/torques can be estimated according to

$$\hat{\mathbf{F}} = \left(\mathbf{J}_g^T(\mathbf{q})\right)^{-1} \hat{\boldsymbol{\tau}}_{ext} \quad (3.3)$$

In order to distinguish between motor torques that cause a motion or a position of the manipulator, and the motor torques induced from external forces/torques, the following notion is used in the context of force estimation. Torques that are needed to hold a position or cause a motion are denoted by  $\boldsymbol{\tau}_\mu$  and torques that result from external forces or torques are denoted by  $\boldsymbol{\tau}_{ext}$ . Furthermore, if friction is present in the joints, torques are needed to overcome the friction. The torques induced by friction are denoted by  $\boldsymbol{\tau}_{fr}$ . The total torque in the joints is then

$$\boldsymbol{\tau} = \boldsymbol{\tau}_\mu + \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_{fr} \quad (3.4)$$

A possible torque estimation method is to calculate the torques in the joints based on the motor currents. This is possible if motor current measurements



are available. The current-torque relationship is proportional in the case of permanent magnet direct current (DC) motors, such that the motor torques  $\boldsymbol{\tau}$  in the motors can be estimated as

$$\hat{\boldsymbol{\tau}} = \mathbf{k}_\tau^T \mathbf{i}_a \quad (3.5)$$

where  $\mathbf{k}_\tau \in \mathbb{R}^n$  is a vector with torque constants and  $\mathbf{i}_a \in \mathbb{R}^n$  is a vector with armature currents from the motors. Note that while this equation holds for DC motors, the current-torque relationship for AC motors is nonlinear and difficult to model. Because the current-torque equations vary between different AC motor types, it is important to know the type, characteristics and parameters of the motor in order to obtain a correct model.

In the following, Equation (3.5) can be used to estimate the external forces and torques acting on the end effector. Substituting  $\hat{\boldsymbol{\tau}}$  in Equation (3.5) with the equivalent from Equation (3.4) yields

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_\mu + \hat{\boldsymbol{\tau}}_{ext} + \hat{\boldsymbol{\tau}}_{fr} = \mathbf{k}_\tau^T \mathbf{i}_a \quad (3.6)$$

In order to estimate the external force caused by  $\boldsymbol{\tau}_{ext}$ , it is necessary to obtain estimates for  $\boldsymbol{\tau}_\mu$  and  $\boldsymbol{\tau}_{fr}$ , and to know the torque constants  $\mathbf{k}_\tau$ , such that

$$\hat{\mathbf{F}} = (\mathbf{J}_g^T(\mathbf{q}))^{-1} \hat{\boldsymbol{\tau}}_{ext} = (\mathbf{J}_g^T(\mathbf{q}))^{-1} (\mathbf{k}_\tau^T \mathbf{i}_a - \hat{\boldsymbol{\tau}}_\mu - \hat{\boldsymbol{\tau}}_{fr}) \quad (3.7)$$

The torques  $\boldsymbol{\tau}_\mu$  related to the position and motion of the manipulator are given by the equations of motion according to Equation (2.64)

$$\hat{\boldsymbol{\tau}}_\mu = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (3.8)$$

where  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  are obtained from measurements.

The torques  $\boldsymbol{\tau}_{fr}$  related to the friction in the joints are often assumed to be negligible or left out as friction is difficult to model. While this assumption can be justifiable for certain types of motors, friction torques can be of high magnitude in harmonic drive AC motors. The Coulomb friction model and viscous friction model are therefore introduced in the next section.

### 3.2.1. Coulomb Friction Model and Viscous Friction Model

The classical model of friction where the friction force is proportional to load, opposite to motion and independent of contact area is known as Coulomb friction [6]. The friction force in the Coulomb model is given by

$$F_{fc} = F_c \text{sgn}(v), \quad v \neq 0 \quad (3.9)$$

### 3. Principles of Force Estimation

---

where  $v$  is the velocity and the Coulomb force  $F_c$  is given by

$$F_c = \rho F_N \quad (3.10)$$

with the friction coefficient  $\rho$  and the load  $F_N$ . For the Coulomb friction model it is assumed that there is no contamination of the contact surfaces, such as lubrication. Such friction is referred to as dry friction.

Viscous friction is present in fluid lubricated contacts between solids. Due to hydrodynamic effects, the friction force of the viscous friction model takes into account both the magnitude of the velocity as well as the direction. The usual linear model is given by

$$F_{fv} = F_v v \quad (3.11)$$

where the viscous friction is proportional to velocity. The constant of proportionality  $F_v$  depends on lubricant viscosity, loading and contact geometry.

The Coulomb and the viscous friction models are used to model the torque friction  $\tau_{fr}$  in the joints, such that

$$\tau_{fr} = \tau_{cou} + \tau_{vis} \quad (3.12)$$

where  $\tau_{cou}$  is the Coulomb friction and  $\tau_{vis}$  is the viscous friction. They can be expressed according to

$$\tau_{cou,i} = \nu_{c,i} \operatorname{sgn}(\dot{q}_i), \quad \dot{q}_i \neq 0, \quad i \in \{1, \dots, n\} \quad (3.13)$$

$$\tau_{vis} = \boldsymbol{\nu}_v^T \dot{\boldsymbol{q}} \quad (3.14)$$

where  $\nu_{c,i}$  are the coulomb friction coefficients and  $\boldsymbol{\nu}_v^T = [\nu_{v,1}, \dots, \nu_{v,n}]$  are the viscous friction coefficients.

A problem arises for the case where  $\dot{q}_i = 0$ , as the Coulomb model in Equation (3.13) is not defined at zero velocity. It is common in simulations to implement the Coulomb friction model according to

$$\tau_{cou,i} = \nu_{c,i} \operatorname{sgn}(\dot{q}_i) = \begin{cases} -\nu_{c,i}, & \dot{q}_i < 0 \\ 0, & \dot{q}_i = 0 \\ \nu_{c,i}, & \dot{q}_i > 0 \end{cases} \quad (3.15)$$

In order to avoid simulation problems caused by frequent zero velocity detection, it is possible to use a dead zone around  $\dot{q}_i$ , such that  $\dot{q}_i = 0$  when  $|\dot{q}_i| \leq \delta$ .

It is obvious that the model in Equation (3.15) does not reflect the physics of friction, as stiction usually causes high friction when  $\dot{q}_i = 0$ . It is therefore recommended to use the Karnopp friction model, which models friction at  $\dot{q}_i = 0$ . However, the friction model used for the remainder of this thesis is the Coulomb friction model according to Equation (3.15), due to simplicity.

### **3.3. Summary**

It is necessary to compute and avoid singular configurations in order to obtain correct force estimates. A simple method of force estimation is based on motor current measurements. Friction modeling is especially important for harmonic drive AC motors. Even though it is recommended to use a Karnopp friction model, the simpler Coulomb friction model is used for the remainder of this thesis.



## 4. Laboratory Equipment: Manipulator, Force Sensor and Accelerometer

The equipment that is used throughout this study, i.e., the UR5 manipulator, the ATI F/T sensor and the Xsens accelerometer, is introduced in Sections 4.1, 4.2 and 4.3. This introduction is followed by a description of the experimental setup and implementations that are necessary to run the ongoing experiments, given in Section 4.4.

### 4.1. The UR5 Manipulator

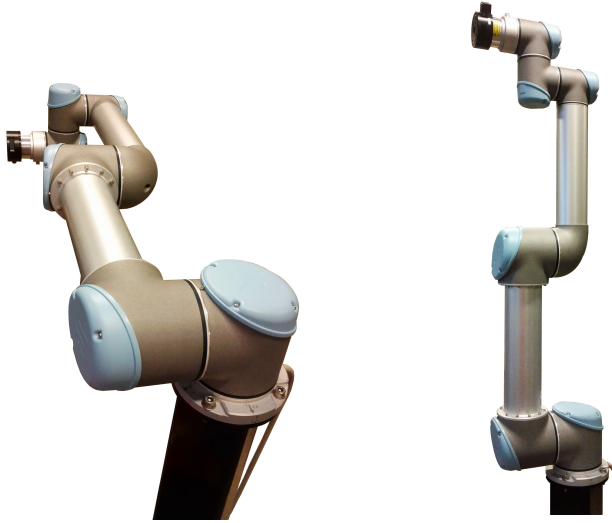
The UR5 is an entry level 6-DOF industrial manipulator manufactured by Universal Robots. With a weight of 18.4 kg it is a lightweight manipulator. It has a reach of 85 cm and a maximal payload of 5 kg and is shown in two positions in Figure 4.1.

The next section gives an overview of the control methods that can be used on the UR5 and lists available parameters and measurements.

#### 4.1.1. Control Methods

The UR5 includes a controller platform with a teaching pendant that allows the user to program the robot using a graphical user interface. This programming interface constraints the options of control to Point-To-Point (PTP) movement in either joint-space or task-space. The default of the PTP movement is that the robot accelerates to the limited velocity, stays there for the maximum time allowed and decelerates to a halt when it reaches the implemented point in space. This results in a trapezoidal velocity trajectory. Alternatively the user can specify a blend radius which gives the robot the freedom to deviate from the original path within that circle around the programmed point. This allows the robot to keep a constant speed and drive through the desired path faster without stopping. The downside of the blends is that a blend radius of 5–10 cm is recommended which impedes a continuous motion with low accuracy [32].

An alternative way to control the robot is to write programs in a scripting language called URScript, which is developed by the manufacturer [31]. The



(a) Position 1:  $\mathbf{q} = [0, 0, 0, 0, 0, 0]^T$       (b) Position 2:  $\mathbf{q} = [0, -\frac{\pi}{2}, 0, -\frac{\pi}{2}, 0, 0]^T$

**Figure 4.1.:** The UR5 manipulator produced by Universal Robots. It is a lightweight manipulator with 18.4 kg, has a reach of 85 cm and a maximal payload of 5 kg. The length of the robot when standing in position two is  $\sim 1.04$  m. The blue caps cover the access to the motors. Directly under the blue caps is mostly air, such that the motors are moved away from the caps within the joint.

programs can be saved directly on the robot controller or commands can be sent via a Transmission Control Protocol (TCP) socket to the robot. These programs are processed in the native high-level controller. It gives the user more options to customize a PTP movement in either joint-space or task-space. It is possible to implement the movement with maximum velocity allowed by the robot controller. The range of joint positions, velocity and acceleration are listed in Table 4.1. Note that even though the manufacturer has documented a maximum acceleration of  $\pm 15 \frac{\text{rad}}{\text{s}^2}$  for the big joints, tests have shown that the big joints reach accelerations up to  $\pm 25 \frac{\text{rad}}{\text{s}^2}$ . This control method has the same characteristics as the previous method, namely that the robot has to come to

a halt between waypoints or that a significant blend radius has to be used to keep a constant velocity.

A third way to control the robot is using the in C programmed Application Programming Interface (C-API) [31]. This enables user coded C-programs to be executed and interact with the controller with a cycle time of 8 ms giving access to the low-level functions of the robot [23]. More precisely, the Universal Robots servo controller can be controlled by either communicating joint velocities or a combination of joint positions, joint velocities and joint accelerations. As compared to the ways presented above to control the UR5, this method is not constrained by a superimposed velocity or acceleration profile and responds to commands quickly with a response delay time of only 12 ms [23].

The preferred control method for research purposes is through the C-API as it gives most access to the control layers. However, the C-API has to be provided by the manufacturer. This was not the case during the period of this study and it is unknown if access will be granted in the future. Therefore the range of possibilities is currently constrained to the teaching pendant and the use of the scripting language URScript. As URScript allows a communication with the robot through an external personal computer (PC), this method is chosen for the remaining part of this thesis.

The next section gives an overview of the provided parameters and measurements of the UR5.

#### **4.1.2. Provided Parameters and Measurements**

Universal Robots provides the user with a few parameters about the UR5. Each joint contains a brushless AC servo motor and a harmonic drive reducer. Useful information about the joints is given in Table 4.1. It lists maximum and minimum joint positions, velocities and accelerations, and gives values named friction and torque constants.

Furthermore, the manufacturer provides the mass and the mass center points for each link, with respect to the coordinate frame of the respective link. The values are listed in Table 4.2.

Additionally, it is possible to obtain real-time signals from the UR5. Table 4.3 lists the data that can be received through the UR5 realtime communication interface. The list is limited to the data that is relevant in this study. Note that merely the joint positions and currents can be assumed to be measurements. Other important values like the joint torques as well as the acceleration and forces at the tool center point are therefore assumed to be modeled values.

**Table 4.1.:** Minimum and maximum values for the joint positions  $\mathbf{q}_{\max}$ , velocities  $\dot{\mathbf{q}}_{\max}$  and accelerations  $\ddot{\mathbf{q}}_{\max}$ , as provided by the manufacturer. Note that even though the manufacturer has mentioned a maximum acceleration of  $\pm 15 \frac{\text{rad}}{\text{s}^2}$  for the big joints, tests have shown that the big joints reach accelerations up to  $\pm 25 \frac{\text{rad}}{\text{s}^2}$ .

	Big joints	Small joints	Notes
$\mathbf{q}_{\max}$	$\pm 2\pi \text{rad}$	$\pm 2\pi \text{rad}$	
$\dot{\mathbf{q}}_{\max}$	$\pm 3.2 \frac{\text{rad}}{\text{s}}$	$\pm 3.2 \frac{\text{rad}}{\text{s}}$	
$\ddot{\mathbf{q}}_{\max}$	$\pm 15 \frac{\text{rad}}{\text{s}^2}$	$\pm 25 \frac{\text{rad}}{\text{s}^2}$	$\pm 25 \frac{\text{rad}}{\text{s}^2}$ for all joints!
$\boldsymbol{\tau}_{\max}$	150 Nm	28 Nm	
static friction	0	0	
dynamic friction	0.11	0.13	
viscous friction	0.4	0.3	
dynamic friction backdrive	0.07	0.08	
viscous friction backdrive	0.6	0.25	
torque constant	0.13	0.14	

The next section presents the force sensor that is attached to the end effector of the UR5.



**Table 4.2.:** Information about the masses  $M_i$  and vectors pointing to the mass center points  $\mathbf{r}_{M_i}^i$  of each link, as provided by the manufacturer. The vectors  $\mathbf{r}_{M_i}^i$  are expressed in the coordinate frame  $o_i x_i y_i z_i$  that is attached to link  $l_i$ .

Link	Mass [kg]	$\mathbf{r}_{M_i}^i$ [mm]
1	$M_1 = 3.7$	$\mathbf{r}_{M_1}^1 = [ 0 \quad , -25.61 \quad , \quad 1.93 ]^T$
2	$M_2 = 8.393$	$\mathbf{r}_{M_2}^2 = [ 212.5 \quad , \quad 0 \quad , \quad 113.36 ]^T$
3	$M_3 = 2.275$	$\mathbf{r}_{M_3}^3 = [ 119.93 \quad , \quad 0 \quad , \quad 26.5 ]^T$
4	$M_4 = 1.219$	$\mathbf{r}_{M_4}^4 = [ 0 \quad , -1.8 \quad , \quad 16.34 ]^T$
5	$M_5 = 1.219$	$\mathbf{r}_{M_5}^5 = [ 0 \quad , \quad 1.8 \quad , \quad 16.34 ]^T$
6	$M_6 = 0.1879$	$\mathbf{r}_{M_6}^6 = [ 0 \quad , \quad 0 \quad , \quad -1.159 ]^T$

**Table 4.3.:** The UR5 realtime communication interface (also known as the Matlab interface) is found at TCP port 30003. The data packages are sent out at a frequency of 125 Hz. Only values relevant to this thesis are listed.

Name	Size in bytes	Description
Message size	4	Total message length in bytes
Time	8	Time elapsed since the controller was started
$\mathbf{q}^d$	48	Target joint positions, $\mathbf{q}^d \in \mathbb{R}^6$
$\dot{\mathbf{q}}^d$	48	Target joint velocity, $\dot{\mathbf{q}}^d \in \mathbb{R}^6$
$\ddot{\mathbf{q}}^d$	48	Target joint accelerations, $\ddot{\mathbf{q}}^d \in \mathbb{R}^6$
$\mathbf{i}^d$	48	Target joint currents, $\mathbf{i}^d \in \mathbb{R}^6$
$\boldsymbol{\tau}^d$	48	Target joint torques, $\boldsymbol{\tau}^d \in \mathbb{R}^6$
$\mathbf{q}$	48	Actual joint positions, $\mathbf{q} \in \mathbb{R}^6$
$\dot{\mathbf{q}}$	48	Actual joint velocities, $\dot{\mathbf{q}} \in \mathbb{R}^6$
$\mathbf{i}$	48	Actual joint currents, $\mathbf{i} \in \mathbb{R}^6$
$\ddot{\mathbf{d}}$	24	Tool X, Y, Z acceleration, $\ddot{\mathbf{d}} \in \mathbb{R}^3$
Unused	120	Unused
$\mathbf{F}$	48	Generalized forces at the tool center point, $\mathbf{F} \in \mathbb{R}^6$
Tool vector	48	Cartesian coordinates of the tool: (X, Y, Z, Rx, Ry, Rz), where Rx, Ry, Rz is a rotation vector representation of the tool orientation, (6 × 1)-vector
Tool speed	48	Speed of the tool given in cartesian coordinates, (6 × 1)-vector

## 4.2. ATI Force/Torque Sensor

The F/T sensor used in the experimental setup is the ATI Gamma F/T Transducer with Net F/T system interface. The sensor measures all six components of force and torque, i.e.,  $F_x$ ,  $F_y$ ,  $F_z$ ,  $n_x$ ,  $n_y$  and  $n_z$ , and is suitable for typical industrial manipulator applications like robotic assembly, grinding and polishing. The transducer uses silicon strain gauges which gives high overload protection while providing a high signal-to-noise ratio. A cylinder shaped attachment comes with the sensor that fits the UR5. The force sensor has a height of 3.3 cm and a radius of 3.7 cm and is shown in Figure 4.2. The attachment has a height of 2 cm and a radius of 3 cm. Both sensor and attachment weigh 460 g together.

The Net F/T system interface connects to Ethernet, Internet Protocol (IP), DeviceNet, and/or Controller Area Network (CAN) bus system. The Net F/T's web browser interface allows to configure and setup the sensor. The system is powered by Power over Ethernet (PoE) or by an external power supply. The Ethernet interface sends high-speed streaming data with up to 7000 Hz using User Datagram Protocol (UDP) packets.

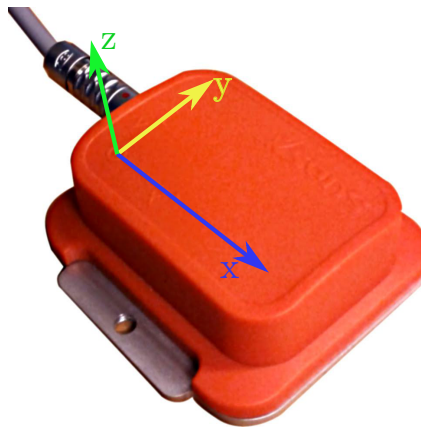


**Figure 4.2.:** The ATI Gamma netft F/T sensor which measures all six components of force and torque, i.e.,  $F_x$ ,  $F_y$ ,  $F_z$ ,  $n_x$ ,  $n_y$  and  $n_z$ .

The next section presents the accelerometer that is attached to the end effector of the UR5.

### 4.3. Xsens Accelerometer

The Xsens Development Kit (MTI-28A53G35) which includes a MTi accelerometer is used in the experimental setup. The accelerometer is a MEMS Inertial Measurement Unit (IMU) providing 3D orientation, 3D inertial data and 3D magnetic field at a maximal frequency of 512 Hz. It has a  $5g \approx 50 m/s^2$  full scale acceleration and a  $300^\circ/s$  full scale rate of turn and can be connected to a PC via Universal Serial Bus (USB). A summary of technical specifications and sensor performance can be found in Table 4.4 and 4.5. The MTi is  $(58 \times 58 \times 22)$  mm small, weighs 50g and is visualized in Figure 4.3.



**Figure 4.3.:** MTi Accelerometer with coordinates.

The MTi Development Kit comes with a MT Software Development Kit (SDK) for Windows. By using the MT SDK, a number of pre-set user scenarios are available for optimizing the EKF routine for different applications. Based on the chosen scenario a sensor fusion algorithm will apply optimized filter settings recommended for the application. The MTi corrects its orientation every sample using the gravity and the earth's magnetic field as reference vectors to compensate for integration drift in the gyroscopes. The Xsens sensor fusion algorithm can cope with short-term magnetic disturbances and lateral accelerations, resulting in a reliable orientation estimate. Additionally, the MTi incorporates a magnetic field mapping routine to correct for hard and soft iron effects.

**Table 4.4.:** Technical Specification for the Xsens 3D Motion Tracker MTi, Attitude and Heading [38].

Static accuracy (roll/pitch)	$< 0.5^\circ$	Dynamic range: Roll/Yaw	$\pm 180^\circ$
Static accuracy (yaw) (1)	$< 1^\circ$	Maximum update rate, onboard processing	256 Hz
Angular resolution (2)	$0.05^\circ$	Maximum update rate, external processing	512 Hz
Dynamic range: Pitch	$\pm 90^\circ$	A/D resolution	16 bits

(1): in homogeneous magnetic environment.

(2):  $1\sigma$  standard deviations of zero-mean angular random walk.

**Table 4.5.:** Sensor performance of the Xsens 3D Motion Tracker MTi [38].

	rate of turn	acceleration	magnetic field
Unit	[deg/s]	[m/s <sup>2</sup> ]	[mGauss]
Dimensions	3 axes	3 axes	3 axes
Full Scale	$\pm 300^\circ/s$	$\pm 50 m/s^2$	$\pm 750$ mGauss
Noise	$0.05^\circ/s/\sqrt{Hz}$	$0.002 m/s^2/\sqrt{Hz}$	0.5 mGauss ( $1\sigma$ )
Bandwidth	40 Hz	30 Hz	10 Hz

A downside to using the Xsens Development Kit is that the user friendly implementations are available only for Windows. In order to use the MTi on Linux, a custom application has to be implemented based on the provided C++ source code of the Communication MT classes. Example C++ application code however helps the user to get started on the implementation [37].

The next section describes the experimental setup and implementations and gives information about the software download.

## 4.4. Setup of the Test Stand and Software Implementation

In order to run the accelerometer, force sensor and manipulator at the same time and to gather data simultaneously, a program is necessary that can handle all those tasks. The program has to establish connection to the devices and receive measurements at a specified frequency. The measurements have to be

saved for further data analysis. Linux is chosen as the operating system. The choice of the programming language depends on the requirements stemming from the hardware. An overview is given in the following section.

### 4.4.1. Software Requirements and Implementation

As mentioned in Section 4.3 the Xsens Accelerometer is connected to the PC via USB and the application has to be written in C++, based on libraries provided by the manufacturer.

The force sensor is connected via Ethernet. By knowing the IP address and port where the force measurements are accessible, sockets can be used to send requests for data. A program that performs this task was already implemented in Python, prior to this study.

Communication with the UR5 controller can be established likewise via Ethernet. As for the force sensor, by knowing the IP address of the robot and the ports that connect to the realtime interface or the robot controller, sockets can be used to request data or send URScript commands to the UR5 controller.

When a request is sent to the devices, the program has to wait for an answer, i.e., the request is a blocking call. In order to gather data from several devices at once, the program has to be parallel.

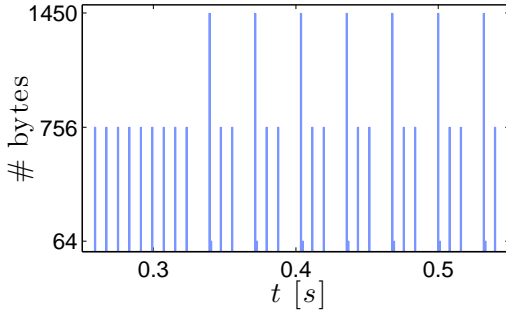
It is practical to use Python to communicate via sockets to the UR5 controller alike to the program for the force sensor. Python includes strong scientific libraries such as Numpy, Scipy and Matplotlib and has become a real open-source alternative to engineering tools such as Matlab and Mathematica [18]. Furthermore, in order to obtain a parallel program, threads are used. Threads are comparatively simple to implement in Python.

Independent components can be used to simplify the implementation, because concurrency problems such as deadlocks are impossible. The main program can be implemented to run each component in parallel.

The implementations for each component are explained in the following sections.

### UR5 Manipulator

The UR5 can be moved by sending URScript commands through TCP port 30001 or 30002, while the realtime interface measurements can be read from port 30003. TCP sockets are used to access the interfaces and two threads are used to implement a parallel program that allows to move the robot at the same time as measurements are received.

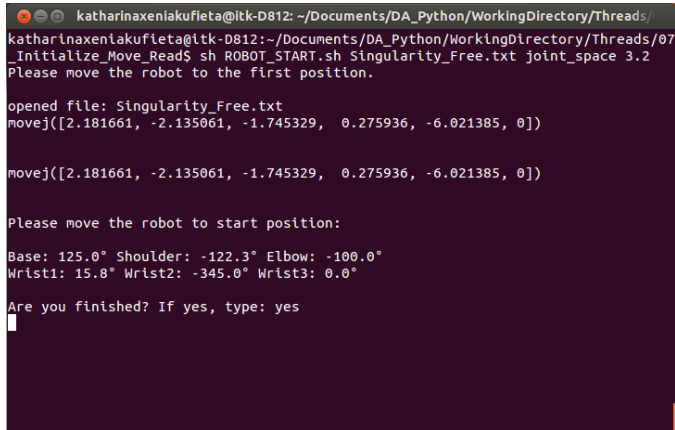


**Figure 4.4.:** Irregularity in the data packages that are received from the UR5 through the realtime communication interface.

**Receive Measurements via Realtime Communication Interface** The realtime communication interface sends out packages at a frequency of 125 Hz. A data package contains a total of 756 bytes. A list of the available data is given in Table 4.3. The requested data is stored in Comma-Separated-Value (CSV) format, together with a time stamp.

By taking a closer look on the saved data, an irregularity is observed in the measurements coming from the UR5 realtime communication interface. Initially, the data arrives at a frequency of 125 Hz as stated by the manufacturer and each package holds 756 bytes. But as soon as the robot starts moving, every third package arrives postponed, together with the newest data package. Furthermore, the newest package is split into two parts, where the first 694 bytes arrive attached to the end of the delayed package and the second part with 62 bytes follows only 1 ms after. The pattern is visualized in Figure 4.4. This problem, when detected, can be solved by changing the program that is reading out the values. By reconstructing the signals, measurements are available every 8 ms when the data is processed offline. In the case of real-time applications, the delay in the data stream has to be taken into consideration.

**Control Robot with URScript** The robot is controlled by sending URScript commands to the UR5 controller. URScript is a Python like language used by the UR5 controller [31]. A typical position command in joint space could be `movej([2.1, -2.1, -1.7, 0.2, -6.23, 0])` and is sent as a string via sockets to the UR5 controller. The values in the command are the desired joint positions,



```
katharinaxeniakufleta@ltk-D812: ~/Documents/DA_Python/WorkingDirectory/Threads/
katharinaxeniakufleta@ltk-D812:~/Documents/DA_Python/WorkingDirectory/Threads/07
_initialize_Move_Read$ sh ROBOT_START.sh Singularity_Free.txt joint_space 3.2
Please move the robot to the first position.

opened file: Singularity_Free.txt
movej([2.181661, -2.135061, -1.745329, 0.275936, -6.021385, 0])

movej([2.181661, -2.135061, -1.745329, 0.275936, -6.021385, 0])

Please move the robot to start position:

Base: 125.0° Shoulder: -122.3° Elbow: -100.0°
Wrist1: 15.8° Wrist2: -345.0° Wrist3: 0.0°

Are you finished? If yes, type: yes
█
```

**Figure 4.5.:** Screenshot of the implemented program.

expressed in radians.

It is observed that the UR5 controller does not consider new commands when the UR5 is still performing the current movement. Instead, the newest command that is sent to the controller after the UR5 came to a halt is considered next. Skipped commands can result in collisions of the robot with itself or the environment, as the initially planned trajectory is not followed correctly. Therefore it is important to wait between commands and to make sure that the robot has enough time to perform each of the movements. The waiting time depends on the distance between two waypoints and must be chosen carefully by the user.

The module that moves the manipulator takes three parameters: A file containing the commands for the robot, a keyword that specifies if the movements are planned in joint-space or task-space and the waiting time between the commands. The user must ensure a collision-free and feasible trajectory defined by way-points. The default waiting time between commands is set to three seconds. When the program is started, it asks the user to move the manipulator into the first position as provided by the file containing the commands. This ensures that the robot does not collide with itself when moving to the start position. Figure 4.5 shows the start of the program.



### Force Sensor

A library that connects to the force sensor by using sockets was already implemented in Python, prior to this study. This program is invoked every 4 ms and the received data is saved together with a time stamp in a CSV file, until terminated.

### Accelerometer

The module that requests and saves data from the accelerometer is based on the CMT Level 3 device library provided by the manufacturer [37].

The accelerometer uses a serial over USB channel. It is initialized by calling provided library functions and specifying the output mode. In this module the output is set to include accelerometer, gyro and magnetometer data. After setting the mode, the data is gathered and written to a CSV file until termination.

#### 4.4.2. Program Synchronization

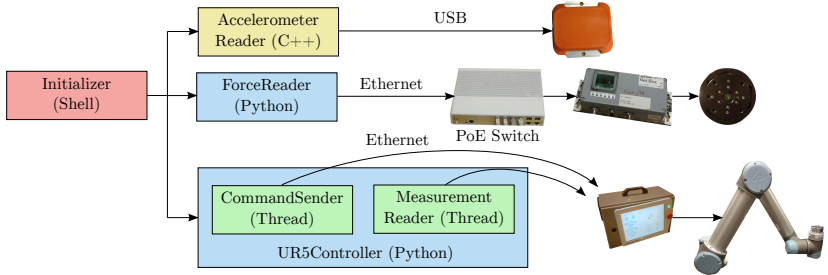
A bash script is used to start all the modules simultaneously. It has to be started in the terminal and takes the three parameters needed to run the robot: the name of the file that contains the position commands for the robot, the commands type and the waiting time between commands. The program asks the user to move the robot manually to the start position by displaying the first position command from the command file. When the user confirms that the robot is at the start position, the bash script starts the accelerometer, force sensor and the two threads for the UR5. The program runs even if no force sensor or accelerometer are connected, but no data for the missing device is recorded. An overview of the entire program structure is given in Figure 4.6.

When the bash script is interrupted it terminates all the concurrent programs.

#### 4.4.3. Data Visualization and Matlab Processing

All measurements are saved in CSV format and stored in a single folder. The data can be plotted immediately using matplotlib, which is a Python 2D plotting library, to view and analyze the results.

A model of the UR5 is implemented in Simulink. It utilizes the data collected from the UR5, accelerometer and force sensor for further offline analysis and simulations, such as the testing of force estimation algorithms. The CSV files containing the data are translated into MAT files, a Matlab data format. Only



**Figure 4.6.:** Program structure of the experimental test stand including accelerometer, force sensor and UR5 manipulator.

the root to the folder that contains the MAT files has to be provided to the Simulink model in order to start the simulation. The time needed to record data directly from the hardware and complete a simulation using the recorded data takes no more than 15 minutes.

Even though the program that records the measurements and runs the robot is a parallel program, an exact synchronization of the measured data cannot be ensured. An automatic interpolation of the data is possible in Matlab while the simulation is running. Therefore it is sufficient to add a timestamp to each measurement and to eventually obtain synchronized data by simulating the Simulink model.

#### 4.4.4. Software Download

Information about how to download the software is given in Appendix B.

### 4.5. Summary

The experiment utilizes three different types of hardware, i.e., the UR5 manipulator, the Xsens MTi Accelerometer and the ATI F/T sensor, which need to be used synchronously. A custom implementation of parallel programs, one for each device, allows to move the robot and to record data from all devices at the same time. UR5 data irregularity can be addressed by reconstructing the distorted data offline. By adding a time stamp to each measurement, the data is automatically synchronized through interpolation in the Matlab simulation.

## 5. UR5 Industrial Manipulator Model

Dynamic models for the UR5 and the end effector acceleration are derived and verified in this chapter. First, the dynamic model of the UR5 as derived in [21] is verified in a closed loop in Section 5.1. To do this an assumption about the underlying control system is made and the control parameters are estimated to complete the closed loop model. Then simulations and experiments are performed to compare the model output with measurements from the robot in Section 5.2. Comparisons between the modeled torques as given by the UR5 manipulator and the modeled torques from [21] suggest that the latter model is inaccurate. New models are therefore derived based on the Euler-Lagrange method with different approaches to model the links of the robot in Section 5.3. The models are verified and compared in Section 5.4 and shown to give significantly better results than the model from [21]. One of the models is chosen for the remainder of the thesis. Eventually, the end effector acceleration is modeled and compared to the measurements from the accelerometer in Section 5.5.

### 5.1. PD Parameter Estimation for the Closed Loop Model

The dynamic model derived by [21] was set as a basis to be verified and used in this thesis. Because direct open loop verification of the model is not possible, the model has to be verified in a closed loop with the measurements that are available from the manipulator. To complete the closed loop model, an assumption about the control system has to be made and the control parameters have to be estimated. It is assumed that a PD controller is used, together with a linearization of the manipulator dynamics. Next, the PD parameters must be estimated to verify if the assumption is correct and to complete the closed loop model. The estimation is based on measurements that are recorded while the UR5 follows a specified trajectory. A list of all the values that are sent from the manipulator is given in Table 4.3. Note that only the joint positions and currents are assumed to be measurements. Other important values like the joint torques as well as the acceleration and forces acting on the end effector are assumed to be modeled values. Nevertheless, those values can be used as additional auxiliary material to verify dynamic models.

**Table 5.1.:** MAE of the modeled positions, velocities and torques for the closed loop with PD control parameters from Equation (C.1) and the dynamic model by [21]. The modeled values are compared to the position and velocity measurements and to the modeled torques stemming from the UR5 manipulator.

Joint	MAE( $\hat{q}_i$ )	MAE( $\hat{\dot{q}}_i$ )	MAE( $\hat{\tau}_i$ )
1	0.0001	0.0024	0.7090
2	0.0002	0.0028	25.8024
3	0.0003	0.0028	14.2721
4	0.0004	0.0070	1.2721
5	0.0003	0.0050	0.2001
6	0.0003	0.0120	0.0333

The measurements that are used for the PD parameter estimation are the joint positions  $\mathbf{q}$  and joint velocities  $\dot{\mathbf{q}}$ , with  $\mathbf{y}$  from Equation (2.81). It is assumed that the PD controller has gain matrices according to Equation (2.73) such that  $\boldsymbol{\kappa} = [\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1] \in \mathbb{R}^{12}$  are unknown and need to be estimated. This leads to an optimization problem as in Equation (2.83). After running the estimation on a random trajectory with different initial parameters to rule out local suboptimal solutions, the estimated parameters are found and listed in Equation (C.1).

The Mean Absolute Errors (MAE) of the modeled positions, velocities and modeled torques are

$$\begin{aligned}
 \text{MAE}(\hat{q}_i) &= \frac{1}{N} \sum_{j=1}^N |q_{i,j} - \hat{q}_{i,j}|, \quad i \in \{1, \dots, 6\} \\
 \text{MAE}(\hat{\dot{q}}_i) &= \frac{1}{N} \sum_{j=1}^N |\dot{q}_{i,j} - \hat{\dot{q}}_{i,j}|, \quad i \in \{1, \dots, 6\} \\
 \text{MAE}(\hat{\tau}_i) &= \frac{1}{N} \sum_{j=1}^N |\tau_{i,j} - \hat{\tau}_{i,j}|, \quad i \in \{1, \dots, 6\}
 \end{aligned} \tag{5.1}$$

respectively, where  $N$  is the number of measurements that are taken. The MAEs are listed in Table 5.1.

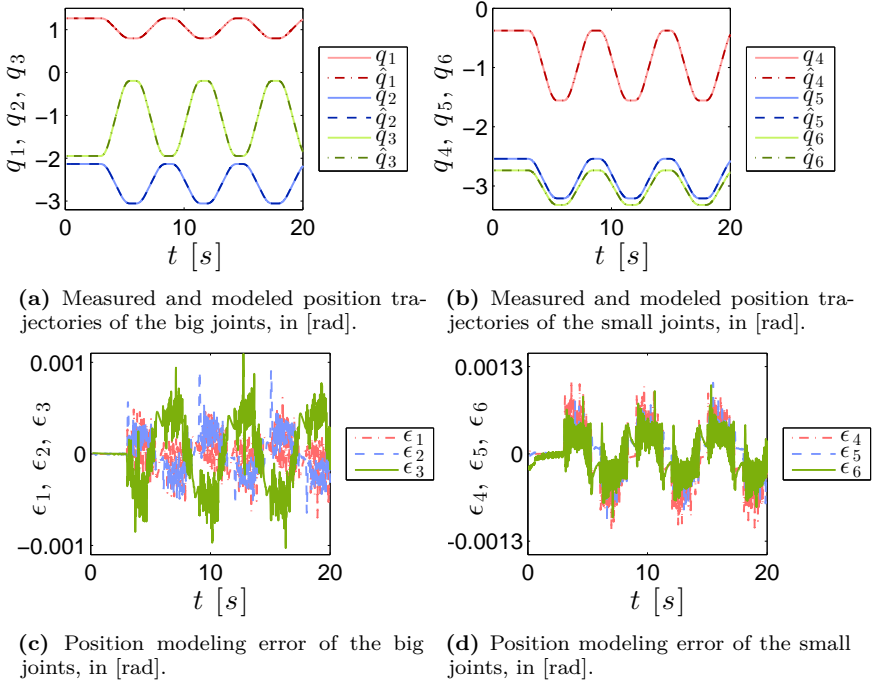
The MAE for the joint positions and velocities can only be verified in a closed loop for a linearized system that equals a double integrator system. Consequently they are, as expected, very small. The torque estimates, however, show a big discrepancy when compared with the torque estimates stemming from the manipulator. Most evident are the MAEs for  $\hat{\tau}_2$  and  $\hat{\tau}_3$  with  $\text{MAE}(\hat{\tau}_2) = 25.8024$  and  $\text{MAE}(\hat{\tau}_3) = 14.2721$ , respectively. This difference and possible reasons are investigated in the next section.

## 5.2. Motivation for a New Dynamic Model

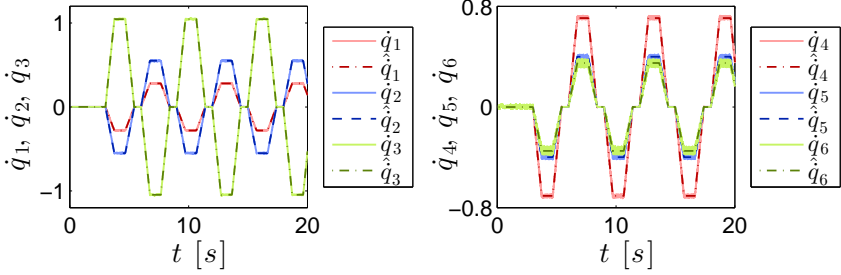
Figures 5.1 and 5.2 show position and velocity trajectories from both UR5 measurements and the model derived from the author [21], in a closed loop with PD parameters from Equation (C.1).

The position and velocity trajectories reflect the MAE values from Table 5.1, as the position and velocity modeling errors are small with absolute boundaries  $|\epsilon| \leq 0.0011$  and  $|\dot{\epsilon}| \leq 0.042$ . Figure 5.3 shows the corresponding torques from the UR5 and the modeled torques. Figure 5.4 shows the difference between the torques from the UR5 and the modeled torques.

The high values for  $\text{MAE}(\hat{\tau}_i)$  in Table 5.1 are reflected in the torque plots as there are evidently big discrepancies between the torques from the UR5 and the modeled torques. Moreover as the mean values for the torque in joint 3 are  $\bar{\tau}_3 = 0.1680$  and  $\bar{\tau}_3 = 14.4607$ , which shows a clear disparity, a closer look at the dynamic equations as derived in [21] imply that the model cannot be correct. Some of the entries of the gravity vector  $\mathbf{g}(\mathbf{q})$  derived by the author allow only values that are close to zero, whilst the impact of gravity on those torques cannot be that small. This is reflected in the torque values  $\boldsymbol{\tau}$  from the UR5, as torques are present in all joints. It can be therefore assumed that the model derived by Universal Robots gives better torque estimates than the model from the thesis [21]. A new model is derived in the next section to investigate if the modeling performance can be improved.

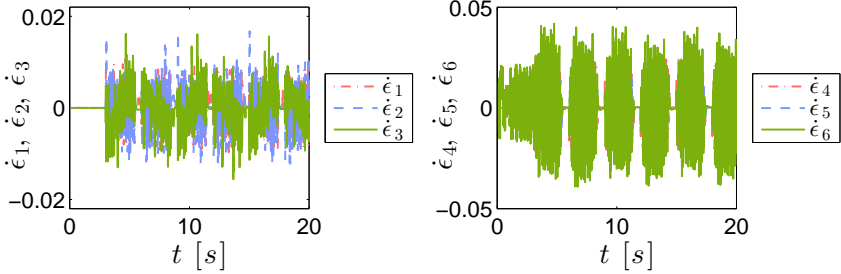


**Figure 5.1.:** Measured and modeled position trajectories for the dynamic model [21] in a closed loop with PD parameters from Equation (C.1).  $q_i$  are the measured positions and  $\hat{q}_i$  are the modeled positions, for  $i \in \{1, \dots, 6\}$ .  $\epsilon_i = q_i - \hat{q}_i$  are the position modeling errors. The big joints are the first three joints with  $i \in \{1, 2, 3\}$  and the small joints are the last three joints with  $i \in \{4, 5, 6\}$ .



(a) Measured and modeled velocities of the big joints, in [rad/s].

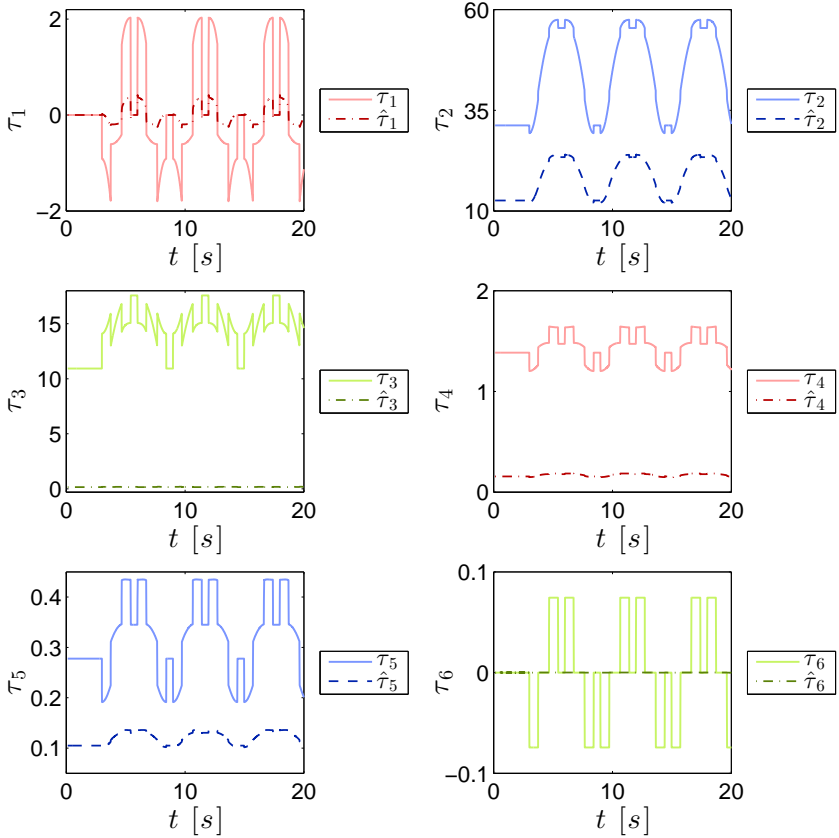
(b) Measured and modeled velocities of the small joints, in [rad/s].



(c) Velocity modeling error of the big joints, in [rad/s].

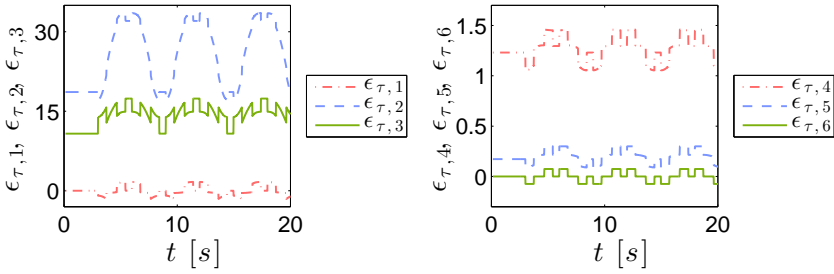
(d) Velocity modeling error of the small joints, in [rad/s].

**Figure 5.2.:** Measured and modeled velocity trajectories for the dynamic model [21] in a closed loop with PD parameters from Equation (C.1).  $\dot{q}_i$  are the measured velocities and  $\hat{q}_i$  are the modeled velocities, for  $i \in \{1, \dots, 6\}$ .  $\dot{\epsilon}_i = \dot{q}_i - \hat{q}_i$  are the velocity modeling errors. The big joints are the first three joints with  $i \in \{1, 2, 3\}$  and the small joints are the last three joints with  $i \in \{4, 5, 6\}$ .



**Figure 5.3.:** Modeled torques of all joints as a result of the applied position and velocity trajectories from Figures 5.1 and 5.2.  $\tau_i$  are the modeled torques as given by the UR5 manipulator, whereas  $\hat{\tau}_i$  are the modeled torques from [21], for  $i \in \{1, \dots, 6\}$ , in [Nm].





**Figure 5.4.:** Modeling difference  $\epsilon_{\tau,i} = \tau_i - \hat{\tau}_i$  between the modeled torques of all joints, for  $i \in \{1, \dots, 6\}$ , in [Nm].  $\tau_i$  are the modeled torques as given by the UR5 manipulator and  $\hat{\tau}_i$  are the torques from [21].

### 5.3. Modeling of the UR5 Manipulator

Having demonstrated in the previous section that the UR5 model derived in [21] has torque modeling errors, new dynamic equations are calculated in this section according to Equations (2.64) to (2.67). Three models are derived based on different link modeling methods. The necessary transformation matrices, Jacobians and inertia tensors are derived in the following section.

#### 5.3.1. Forward Kinematics

Figure 5.5 shows a sketch of the UR5 manipulator with its joints and links. The manipulator has seven links  $l_i : i \in \{0, \dots, 6\}$  and six revolute joints  $j_i : i \in \{1, \dots, 6\}$ . Each revolute joint has one DOF, so the UR5 has a total of six DOF. The first step in order to derive the forward kinematics is to find the DH parameters.

#### DH Parameters

The DH parameters for the UR5 are derived according to the DH convention as presented in Section 2.2.2. The first step is to make a sketch of the manipulator with its joints and links, see Figure 5.5. The measurements of the size of the links are given by the manufacturer and were verified directly on the manipulator. Next, the coordinate frames  $o_i x_i y_i z_i, \forall i \in \{0, \dots, 6\}$ , are assigned based on the sketch and by complying with the DH convention. Figure 5.6 visualizes the assigned coordinate frames. Note that coordinate frame  $o_2 x_2 y_2 z_2$  is not lying on the second link as this minimizes the number of non-zero DH parameters and makes the subsequent transformation matrices neat.

Eventually the DH parameters are extracted using the assigned coordinate frames and the rules of the DH convention. They are visualized in Figure 5.6 and summarized in Table 5.2.

#### Transformation Matrices

By inserting the parameters from Table 5.2 into Equation (2.19) the transformation matrices  $\mathbf{A}_1, \dots, \mathbf{A}_6$  are obtained. They are

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{R}_1^0 & \mathbf{o}_1^0 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_1} & 0 & s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & 0.08916 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2a)$$

**Table 5.2.:** DH parameters and their values for the UR5 manipulator. All joints are revolute joints.

(a) DH parameters overview.					(b) DH parameter values.			
Link $i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$	Link $i$	$d_i$ [mm]	$a_i$ [mm]	$\alpha_i$ [rad]
1	$\theta_1^*$	$d_1$	0	$\alpha_1$	1	89.16	0	$\frac{\pi}{2}$
2	$\theta_2^*$	0	$a_2$	0	2	0	-425	0
3	$\theta_3^*$	0	$a_3$	0	3	0	-392.25	0
4	$\theta_4^*$	$d_4$	0	$\alpha_4$	4	109.15	0	$\frac{\pi}{2}$
5	$\theta_5^*$	$d_5$	0	$\alpha_5$	5	94.65	0	$-\frac{\pi}{2}$
6	$\theta_6^*$	$d_6$	0	0	6	82.3	0	0

\* joint variable

$$A_2 = \begin{bmatrix} \mathbf{R}_2^1 & \mathbf{o}_2^1 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & -0.425 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & -0.425 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2b)$$

$$A_3 = \begin{bmatrix} \mathbf{R}_3^2 & \mathbf{o}_3^2 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & -0.392 c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & -0.392 s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2c)$$

$$A_4 = \begin{bmatrix} \mathbf{R}_4^3 & \mathbf{o}_4^3 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & 0.1092 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2d)$$

$$A_5 = \begin{bmatrix} \mathbf{R}_5^4 & \mathbf{o}_5^4 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & 0.0947 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2e)$$

$$A_6 = \begin{bmatrix} \mathbf{R}_6^5 & \mathbf{o}_6^5 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0.0823 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2f)$$

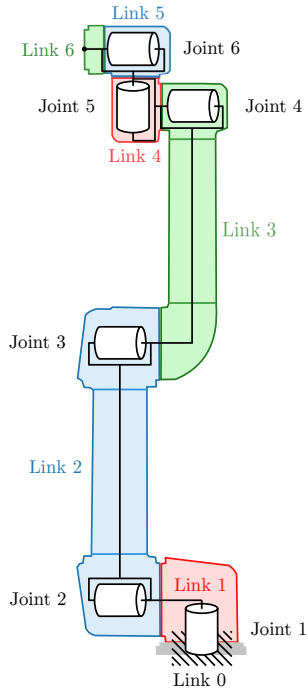
where the parameters  $d_i$  and  $a_i$  were inserted expressed in meters.

The forward kinematics are then described by the transformation matrix  $\mathbf{T}_6^0$  that describes the end effector position and orientation in terms of the base frame  $o_0x_0y_0z_0$  and the joint positions  $\theta_1, \dots, \theta_6$ . It is obtained by substituting Equations (5.2) into Equation (2.18) and results in

$$\mathbf{T}_6^0 = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \begin{bmatrix} \mathbf{R}_6^0 & \mathbf{o}_6^0 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

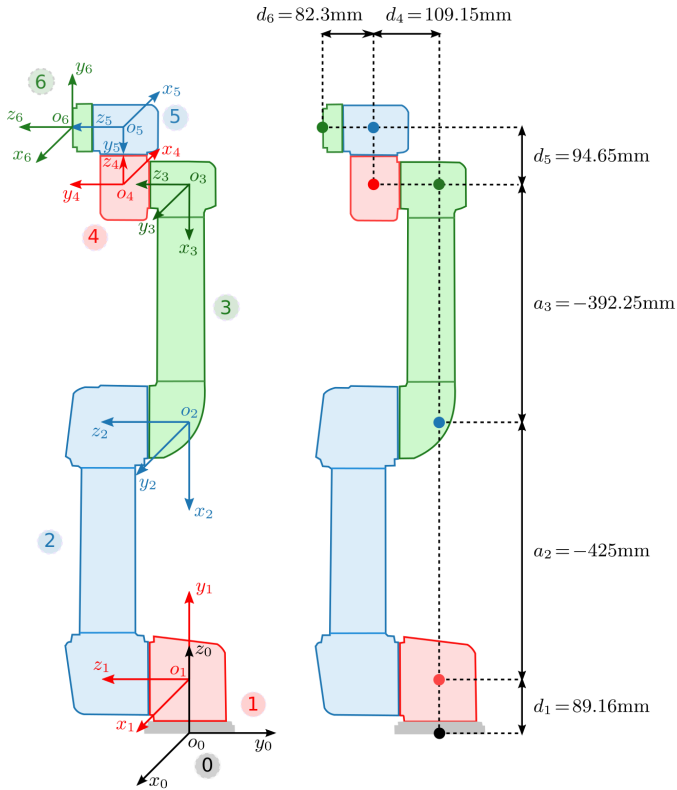
The resulting transformation matrix  $\mathbf{T}_6^0$  as well as the matrices in the following sections are omitted due to their size.

In the following section the velocity kinematics are derived including the geometric and analytical Jacobians.



**Figure 5.5.:** Sketch of the outer shape of the UR5 including its joints and links. It has seven links and six revolute joints. Each joint has a single DOF, so the UR5 has a total of six DOF.

## 5. UR5 Industrial Manipulator Model



**Figure 5.6.:** Left: Sketch of the coordinate frames that were assigned to each link of the UR5 manipulator according to the DH convention. Right: Illustration of the resulting DH parameters.

### 5.3.2. Velocity Kinematics

#### Geometric Jacobian

To calculate the angular part of the geometric Jacobian  $\mathbf{J}_g$ , namely  $\mathbf{J}_{g,\omega}$ , the axes from  $\mathbf{z}_0^0$  to  $\mathbf{z}_5^0$  are needed. They are obtained according to Equation (2.28) and result in

$$\begin{aligned}
 \mathbf{z}_0^0 &= \mathbf{k} \\
 \mathbf{z}_1^0 &= \mathbf{R}_1^0 \mathbf{k} \\
 \mathbf{z}_2^0 &= \mathbf{R}_2^0 \mathbf{k} = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{k} \\
 \mathbf{z}_3^0 &= \mathbf{R}_3^0 \mathbf{k} = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{R}_3^2 \mathbf{k} \\
 \mathbf{z}_4^0 &= \mathbf{R}_4^0 \mathbf{k} = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{R}_3^2 \mathbf{R}_4^3 \mathbf{k} \\
 \mathbf{z}_5^0 &= \mathbf{R}_5^0 \mathbf{k} = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{R}_3^2 \mathbf{R}_4^3 \mathbf{R}_5^4 \mathbf{k}
 \end{aligned} \tag{5.4}$$

where  $\mathbf{k} = [0, 0, 1]^T$  and where all  $\mathbf{R}_{i+1}^i$ , for  $i \in \{0, \dots, 4\}$ , are given in Equation (5.2). The linear part of the Jacobian, namely  $\mathbf{J}_{g,v}$ , is obtained as described in Equation (2.29). The geometric Jacobian is then

$$\mathbf{J}_g = \begin{bmatrix} \mathbf{J}_{g,v} \\ \mathbf{J}_{g,\omega} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{o}_6^0}{\partial q_1} & \frac{\partial \mathbf{o}_6^0}{\partial q_2} & \frac{\partial \mathbf{o}_6^0}{\partial q_3} & \frac{\partial \mathbf{o}_6^0}{\partial q_4} & \frac{\partial \mathbf{o}_6^0}{\partial q_5} & \frac{\partial \mathbf{o}_6^0}{\partial q_6} \\ \mathbf{z}_0^0 & \mathbf{z}_1^0 & \mathbf{z}_2^0 & \mathbf{z}_3^0 & \mathbf{z}_4^0 & \mathbf{z}_5^0 \end{bmatrix} \tag{5.5}$$

where  $\mathbf{o}_6^0 \in \mathbb{R}^3$  is the vector in the 4<sup>th</sup> row of  $\mathbf{T}_6^0$  in Equation (5.3).

#### Analytical Jacobian

The chosen minimal representation for the end effector orientation are the Euler angles  $\phi = [\varphi, \vartheta, \psi]^T$ . Because all the previous calculations were performed with a non-minimalistic orientation representation  $\omega$ , the Euler angles  $\phi$  must be retrieved from the rotation matrix  $\mathbf{R}_6^0$  by using Equations (2.4) to (2.7). The end effector position  $\mathbf{X}$  from Equation (2.30) is then defined and the analytical Jacobian  $\mathbf{J}_a(q)$  can be calculated by inserting the calculated Euler Angles  $\phi(q)$  and the calculated geometric Jacobian  $\mathbf{J}_g(q)$  into Equations (2.33) and (2.34).

#### Jacobian for the Mass Center Points of the Links

The manufacturer provides the position of the mass center points of each link. It is expressed in a frame that is attached to the respective link and is denoted by a vector  $\mathbf{r}_{Mj}^j$ . The provided mass center points are listed in Table 4.2.

## 5. UR5 Industrial Manipulator Model

---

To calculate the equations of dynamics according to Section 2.2.5 it is necessary to transform the vectors into the inertial frame, i.e.,  $\mathbf{r}_{Mj}^0$ . This is done by calculating

$$\begin{aligned}
 \mathbf{r}_{M1}^0 &= \mathbf{o}_1^0 + \mathbf{R}_1^0 \mathbf{r}_{M1}^1 \\
 \mathbf{r}_{M2}^0 &= \mathbf{o}_2^0 + \mathbf{R}_2^0 \mathbf{r}_{M2}^2 \\
 \mathbf{r}_{M3}^0 &= \mathbf{o}_3^0 + \mathbf{R}_3^0 \mathbf{r}_{M3}^3 \\
 \mathbf{r}_{M4}^0 &= \mathbf{o}_4^0 + \mathbf{R}_4^0 \mathbf{r}_{M4}^4 \\
 \mathbf{r}_{M5}^0 &= \mathbf{o}_5^0 + \mathbf{R}_5^0 \mathbf{r}_{M5}^5 \\
 \mathbf{r}_{M6}^0 &= \mathbf{o}_6^0 + \mathbf{R}_6^0 \mathbf{r}_{M6}^6
 \end{aligned} \tag{5.6}$$

where  $\mathbf{o}_j^0$  and  $\mathbf{R}_j^0$  are obtained by extracting  $\mathbf{o}_{i+1}^i$  and  $\mathbf{R}_{i+1}^i$ , for  $i \in \{0, \dots, j-1\}$ , from Equations (5.2) and inserting them into Equations (2.14) and (2.15), respectively. Note that  $\mathbf{d}_j^i = \mathbf{o}_j^i$  in this case. The six Jacobians  $\mathbf{J}_{Mj}$ ,  $j \in \{1, \dots, 6\}$ , one for each mass center point in the inertial frame, are then calculated by substituting Equations (5.4) and (5.6) into Equation (2.40), where  $\mathbf{r}_{Pj}^0 = \mathbf{r}_{Mj}^0$ .

### 5.3.3. Inertia Tensors

As mentioned in Section 5.3.2, the mass center points for each link are provided by the manufacturer. Their positions are given by vectors in the coordinate frames of their respective link. The vectors are listed in Table 4.2. The mass center points are visualized in Figure 5.7a.

To calculate the equations of dynamics and more specifically the kinetic energy of the UR5, the inertia tensors must be calculated in the body attached frame of their respective links. From Figure 5.7a it becomes clear that the links can be approximated with cylinders but that none of the mass center points are located at the centroids of those cylinders.

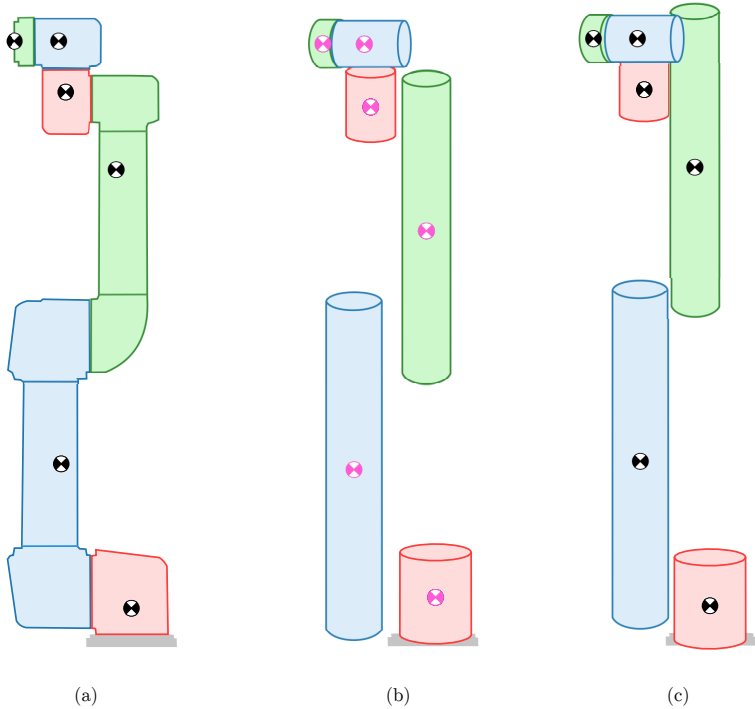
### Assumption of Constant Mass Density

It is common to assume that the links are approximated by cylinders with homogeneous density and that the mass center points are at the centroid of each cylinder [15, 21]. This case is illustrated in Figure 5.7b. The resultant calculations of the inertia tensors are in this case simplified to the Equations (2.47) and (2.48). The mass, height and radius of the cylinders are found from measurements taken on the UR5 as well as from data given by the manufacturer



and are listed in Table C.2. This approximation leads to new mass center points which are pointed to by the vectors  $\mathbf{r}_{Mi,A}^i$ . They are listed together with the original mass center points, provided by the manufacturer and pointed to by the vectors  $\mathbf{r}_{Mi}^i$ , in Table 5.4. Hereafter this model will be referenced to as Model A.

Figure 5.7c shows a second model that was used for comparison. The same cylinders as listed in Table C.2 and visualized in Figure 5.7b are shifted such that their centroids are located on the mass center points provided by the manufacturer. Hereafter this model will be referenced to as Model B.



**Figure 5.7.:** (a) Original mass center points as provided by the manufacturer. They are pointed to by the vectors  $r_{Mi}^i$ . (b) Model A: Approximation of the UR5 links by using cylinders. This leads to new mass center points pointed to by the vectors  $r_{Mi,A}^i$ . They are recolored in pink. (c) Model B: The cylinders from Model A are rearranged to match the mass center points provided by the manufacturer as closely as possible. They are pointed to by the vectors  $r_{Mi,B}^i$ .

**Table 5.4.:** Masses and location of mass center points for each link. The position is given as a vector which is expressed in the to the link attached coordinate frame. The mass center points provided by the manufacturer are pointed to by the vectors  $\mathbf{r}_{Mi}^i$ . The mass center points for Model A and B are pointed to by  $\mathbf{r}_{Mi,A}^i$  and  $\mathbf{r}_{Mi,B}^i$ , respectively. All distances are given in [mm].

Link	Weight $M_i$ [kg]	Location of mass center point $m_{ci}$ in coordinate frame $i$ , [mm]
1	3.7	$\mathbf{r}_{M1}^1 = [ 0, -25.61, 1.93]^T$ $\mathbf{r}_{M1,A}^1 = [ 0, 0, 0 ]^T$ $\mathbf{r}_{M1,B}^1 = [ 0, -25.61, 0 ]^T$
2	8.393	$\mathbf{r}_{M2}^2 = [ 212.5, 0, 113.36]^T$ $\mathbf{r}_{M2,A}^2 = [ 212.5, 0, 135 ]^T$ $\mathbf{r}_{M2,B}^2 = r_{c2,2}$
3	2.275	$\mathbf{r}_{M3}^3 = [ 119.93, 0, 26.5]^T$ $\mathbf{r}_{M3,A}^3 = [ 206.25, 0, 17.5]^T$ $\mathbf{r}_{M3,B}^3 = r_{c3,3}$
4	1.219	$\mathbf{r}_{M4}^4 = [ 0, -1.8, 16.34]^T$ $\mathbf{r}_{M4,A}^4 = [ 0, 0, -2.5 ]^T$ $\mathbf{r}_{M4,B}^4 = [ 0, 0, 16.34]^T$
5	1.219	$\mathbf{r}_{M5}^5 = [ 0, 1.8, 16.34]^T$ $\mathbf{r}_{M5,A}^5 = [ 0, 0, -2.5 ]^T$ $\mathbf{r}_{M5,B}^5 = [ 0, 0, 16.34]^T$
6	0.1879	$\mathbf{r}_{M6}^6 = [ 0, 0, -1.159]^T$ $\mathbf{r}_{M6,A}^6 = [ 0, 0, -17.5 ]^T$ $\mathbf{r}_{M6,B}^6 = [ 0, 0, -17.5 ]^T$

### Consideration of Varying Mass Density

As mentioned above, the mass center points provided by the manufacturer are not aligned with the centroids of the cylinders that are used to approximate the links. This means that the mass is not constantly distributed in the links. By taking a closer look at the UR5 manipulator it becomes clear that the mass distribution of the links is affected by the location of the motors in the joints. By knowing the approximate location and weight of the motors it is possible to calculate the inertia tensors by splitting the links into several bodies as suggested in the method presented in Section 2.2.4.

As a first step observations were made about the UR5 manipulator. It is known from the manufacturer that two types of motors with different size and mass are used in the UR5 manipulator. The big motors are used for the first three joints whereas the small motors are used for the last three joints. This allows the following observations:

- Link 1 contains merely a big motor. It has a mass of 3.7 kg.
- Link 2 contains two big motors. It is known from link 1 that each of the motors weighs 3.7 kg. This leaves  $8.39 - 2 \cdot 3.7 = 0.99$  kg for the middle part of link 2.
- Link 4 and 5 each contain one small motor. Both have a mass of 1.219 kg.
- Link 3 contains a small motor which is known to have a mass of 1.219 kg. This leaves  $2.275 - 1.219 = 1.056$  kg for the rest of link 3.

When opening the housing of the motors by taking off the blue caps, which are visible in Figure 4.1, it becomes clear that the motors are shifted away from the blue cap within the link, which explains the shifted mass center points in links 1, 4, 5 and to a certain extent also the shifted mass center points in links 2 and 3. This justifies a disassembly of links 1, 4, and 5 into two cylinders with homogeneous mass density each, where the masses  $m_{1,cyl_1}$ ,  $m_{1,cyl_2}$ ,  $m_{4,cyl_1}$ ,  $m_{4,cyl_2}$ ,  $m_{5,cyl_1}$ ,  $m_{5,cyl_2}$  are chosen heuristically and the heights  $h_{1,cyl_1}$ ,  $h_{1,cyl_2}$ ,  $h_{4,cyl_1}$ ,  $h_{4,cyl_2}$ ,  $h_{5,cyl_1}$ ,  $h_{5,cyl_2}$  are calculated through Equations (2.54) and (2.55). All of the obtained values are listed in Table C.3. The inertia tensors can then be calculated according to the parallel axis theorem, see Equations (2.45) and (2.46), and Equations (2.56), (2.57) and (2.58), where  $\sigma_{p_{m_i,j}}^{PM_i}$  are listed in Table C.4.

Next, link 2 is disassembled into 3 parts, where the upper and lower cylinder are the big motors and therefore equal to the complete cylinder of link 1, including height, radius, mass and mass center point. The middle part is modeled by

a single cylinder where height, radius, mass and mass center point are likewise known from measurements and previous calculations. As a result there are no open equations. Instead, those equations are used to verify if the hitherto existing assumptions and values are correct. Indeed, the common mass center point for the three prescribed cylinders is equal to the mass center point provided by the manufacturer for link 2. Finally, as the two big motors were not split into two cylinders yet, this is done by using the results of link 1. Eventually link 2 is disassembled into five cylinders with homogeneous mass density and a known height, radius, mass and mass center point. All the values that are necessary to calculate the inertia tensor for link 2 are listed in Table C.3 and C.4.

Link 3 is disassembled into 3 parts, where the uppermost part is equal to link 4 and 5, as it holds a small motor. The middle part is similar to the middle part of link 2, as it is has the same height and material and only the radius is smaller, as  $r_{2,cyl_2} = 43$  mm and  $r_{3,cyl_2} = 37.6$  mm. This suggests that the mass for the middle part of link 3 is  $< m_{2,cyl_2} = 0.993$  kg. The mass for the middle and lowest part of link 3 are then chosen heuristically and are listed in Table C.3. By extending the method as described in Section 2.2.4 link 3 is split into a total of five cylinders with constant mass density. The data necessary for the inertia tensor calculation is listed in Table C.3 and C.4.

Link 6 is the solid metal attachment for end effectors. Even though the mass center point provided by the manufacturer is not aligned with the centroid, this assumption is made here. Link 6 is therefore modeled by a single cylinder with a constant mass density. The data necessary to calculate the inertia tensor is listed in Table C.3 and C.4.

Figure 5.8b shows the divided cylinders and their respective inertia tensors.

### Comparison of Resulting Inertia Tensors

The approximated inertia tensors for the links of the manipulator are equal in Model A and B, as the same cylinders are used in both models. They are

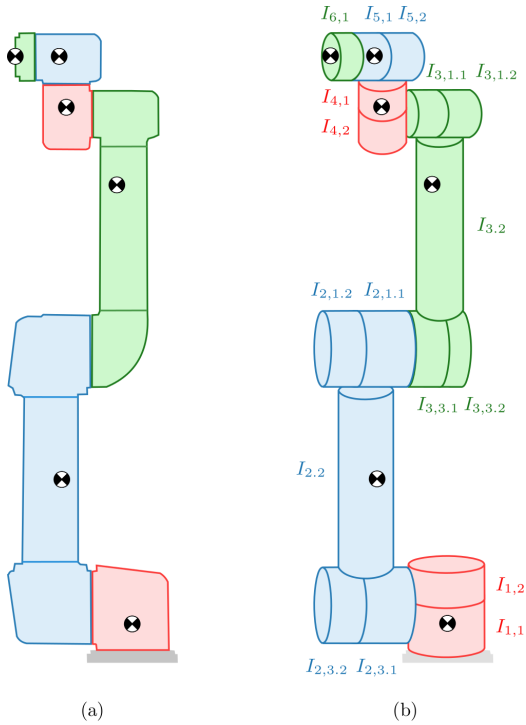
$$\begin{aligned}
 \mathbf{I}_1 &= \begin{bmatrix} 0.0084 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0084 \end{bmatrix} & \mathbf{I}_2 &= \begin{bmatrix} 0.0078 & 0 & 0 \\ 0 & 0.2100 & 0 \\ 0 & 0 & 0.2100 \end{bmatrix} \\
 \mathbf{I}_3 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0462 & 0 \\ 0 & 0 & 0.0462 \end{bmatrix} & \mathbf{I}_4 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0016 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \\
 \mathbf{I}_5 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0016 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} & \mathbf{I}_6 &= \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}
 \end{aligned} \tag{5.7}$$

The inertia tensors for Model C are

$$\begin{aligned}
 \mathbf{I}_1 &= \begin{bmatrix} 0.0067 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} & \mathbf{I}_2 &= \begin{bmatrix} 0.0149 & 0 & 0 \\ 0 & 0.3564 & 0 \\ 0 & 0 & 0.3553 \end{bmatrix} \\
 \mathbf{I}_3 &= \begin{bmatrix} 0.0025 & 0 & 0.0034 \\ 0 & 0.0551 & 0 \\ 0.0034 & 0 & 0.0546 \end{bmatrix} & \mathbf{I}_4 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \\
 \mathbf{I}_5 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} & \mathbf{I}_6 &= \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}
 \end{aligned} \tag{5.8}$$

The values for the moments and products of inertia are small in all models. This is due to the low weight of the links of the UR5. The moments of inertia in link 2 and 3 are bigger in Model C than in Model A and B. Products of inertia are present only in the third inertia tensor of Model C, but not in Model A and B. The moments of inertia in Model C are equal or less than those in Model A and B for the remaining links 1, 4, 5 and 6. The reason for this is that the mass in those links is modeled more closely around the mass center points in Model C than in Model A and B.

Simulations and a verification of the new UR5 manipulator models are performed in the next section in order to find the best dynamic model.



**Figure 5.8.:** (a) Original mass center points as provided by the manufacturer. (b) Model C: Reconstruction of the original mass center points. This is achieved by modeling each link with multiple cylinders of constant mass density. The inertia tensors are written next to their respective cylinders.

## 5.4. Simulation and Verification of the New UR5 Manipulator Models

Three new dynamic models were derived in the previous section. The differences stem from different approaches to calculate the inertia tensors of the links.

The first approach was to model each link of the robot with a cylinder and to assume that the mass center points are equal to the centroids. This approach neglects the information about the mass center points as given by the manufacturer and relies on the newly derived mass center points instead. The result is Model A and is visualized in Figure 5.7b.

The second approach is used only for comparison. It uses the cylinders as they were derived for Model A, but moves the cylinders such that their mass center points are at the positions provided by the manufacturer. The idea is to investigate the effect of the accuracy of the modeled mass center points on the manipulator dynamics. This case results in Model B and is visualized in Figure 5.7c.

The final approach is to model each link with several cylinders in order to obtain the right link shape and mass distribution while reconstructing the mass center point to the position as provided by the manufacturer. It is believed that this model reflects the manipulator dynamics with the highest accuracy. This approach results in Model C and is visualized in Figure 5.8b.

In order to verify Model A, B and C in a closed loop the PD control parameters have to be estimated for each model separately. The parameters are calculated in the same manner as previously shown in Section 5.1 and are listed in Equations (C.2), (C.3) and (C.4) for Model A, B and C, respectively.

The performance of the three models are compared based on the same trajectories as used in Section 5.2. The MAE of the modeled positions, velocities and torques are listed in Tables 5.5 and 5.6.

The MAE for the joint positions and velocities are as before very small and observed differences in the MAE of the modeled joint velocities between the different models are not significant. The torque estimates from all three new models give better results than the torque estimates from [21]. Especially the MAE for the torque in joints two and three drop from  $\text{MAE}(\hat{\tau}_2) = 25.8024$  and  $\text{MAE}(\hat{\tau}_3) = 14.2721$  for model [21] to  $\text{MAE}(\hat{\tau}_2) = 0.5860$  and  $\text{MAE}(\hat{\tau}_3) = 1.2439$  for Model C. The comparison of the torque MAE between the models A, B, and C, shows that Model A has as expected higher torque modeling errors than Model B and C. Surprisingly, the torque modeling errors from Model B and C are similar and none of the two models outperforms the other. Those



**Table 5.5.:** Comparison of MAE of modeled positions and velocities for Model A, B, and C. Smallest errors are colored in green and largest errors are colored in red.

Joint	MAE( $\hat{q}_i$ )			MAE( $\hat{\dot{q}}_i$ )		
	Model A	Model B	Model C	Model A	Model B	Model C
1	0.0001	0.0001	0.0001	0.0028	0.0028	0.0026
2	0.0002	0.0002	0.0002	0.0035	0.0035	0.0031
3	0.0004	0.0004	0.0004	0.0033	0.0033	0.0031
4	0.0005	0.0004	0.0004	0.0111	0.0110	0.0085
5	0.0002	0.0002	0.0002	0.0063	0.0063	0.0056
6	0.0003	0.0003	0.0003	0.0148	0.0148	0.0129

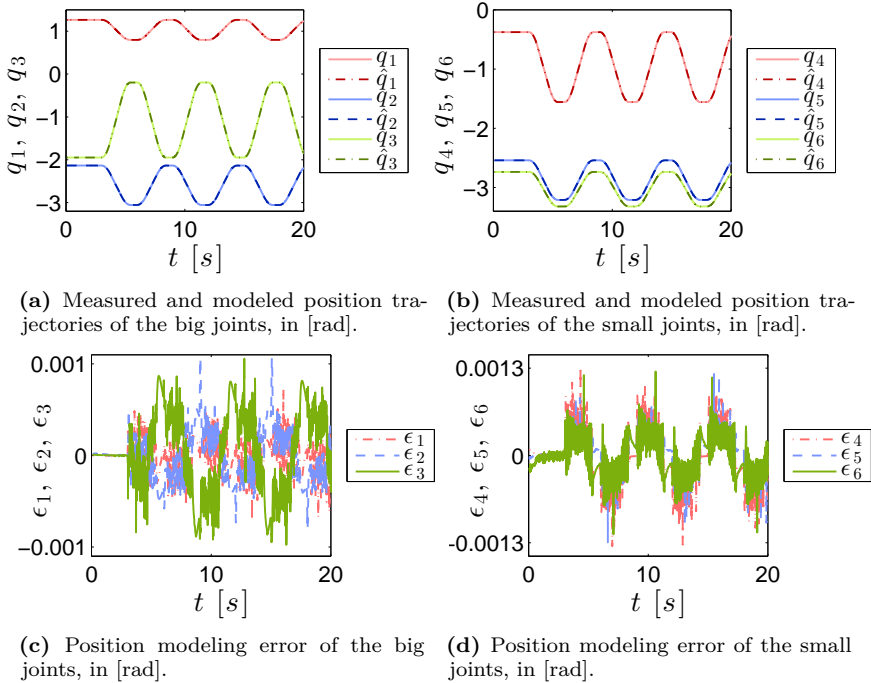
two models use the same mass center points for the links, but different inertia tensors. However, the inertia tensors, given in Equations (5.7) and (5.8), do not differ much, as the masses of the links are small, which is a possible reason for the similar modeling performance. Again, it is important to note that the performance comparison is made based on the torque estimates from the UR5 and not based on measurements. This comparison can therefore not be taken as an absolute performance measure.

Finally, Model C is chosen for the rest of the thesis. As it is based on the most accurate approximation of the links and inertia tensors it can be assumed that it better reflects the real physical system than the other models. Figures 5.9 and 5.10 show the position and velocity trajectories that were used to compare the models, along with the estimated positions and velocities from Model C. Figure 5.11 shows the torques that result from the applied trajectories, along with the modeled torques from Model C. Figure 5.12 shows the difference between the modeled torques.

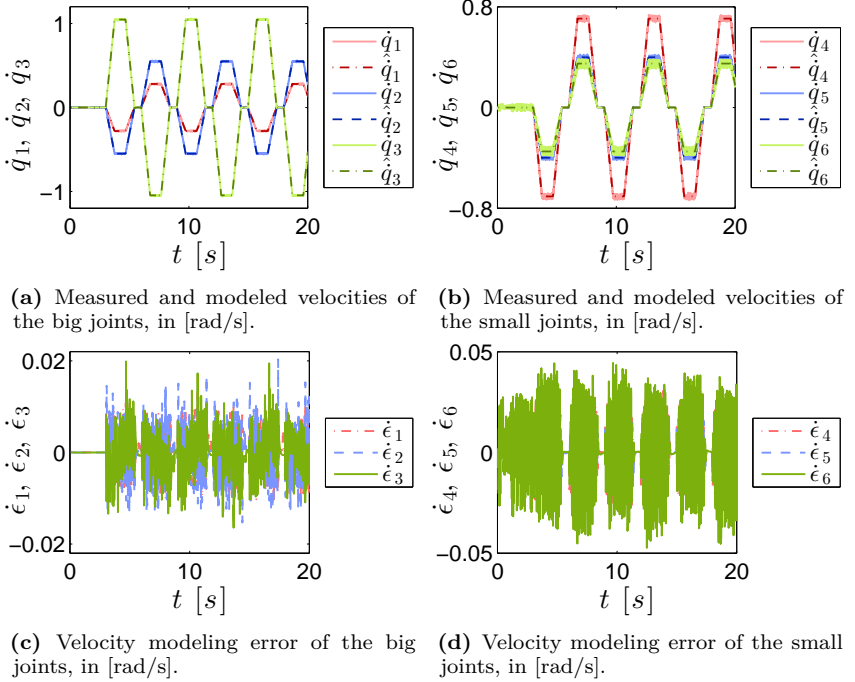
The end effector accelerations are modeled in the next section and are compared to the accelerometer measurements.

**Table 5.6.:** Comparison of MAE of modeled torques for Model A, B, and C. Smallest errors are colored in green and largest errors are colored in red.

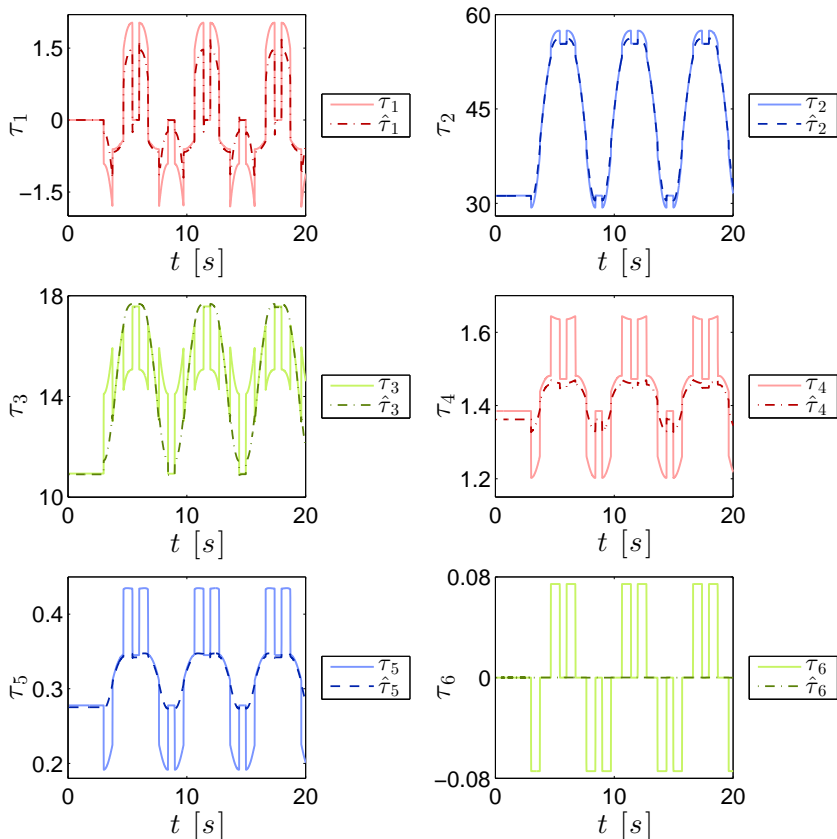
Joint	MAE( $\hat{\tau}_i$ )		
	Model A	Model B	Model C
1	0.3227	0.2973	0.2959
2	1.8140	0.6082	0.5860
3	1.9529	1.1918	1.2439
4	0.2119	0.0808	0.0837
5	0.2387	0.0393	0.0411
6	0.0333	0.0333	0.0348



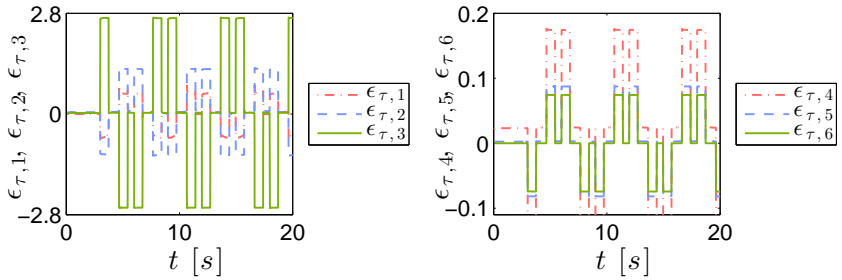
**Figure 5.9.:** Measured and modeled position trajectories for Model C in a closed loop with PD parameters as given in Equation (C.4).  $q_i$  are the measured positions and  $\hat{q}_i$  are the modeled positions, for  $i \in \{1, \dots, 6\}$ .  $\epsilon_i = q_i - \hat{q}_i$  are the position modeling errors. The big joints are the first three joints with  $i \in \{1, 2, 3\}$  and the small joints are the last three joints with  $i \in \{4, 5, 6\}$ .



**Figure 5.10.:** Measured and modeled velocity trajectories for Model C in a closed loop with PD parameters as given in Equation (C.4).  $\dot{q}_i$  are the measured velocities and  $\hat{\dot{q}}_i$  are the modeled velocities, for  $i \in \{1, \dots, 6\}$ .  $\dot{\epsilon}_i = \dot{q}_i - \hat{\dot{q}}_i$  are the velocity modeling errors. The big joints are the first three joints with  $i \in \{1, 2, 3\}$  and the small joints are the last three joints with  $i \in \{4, 5, 6\}$ .

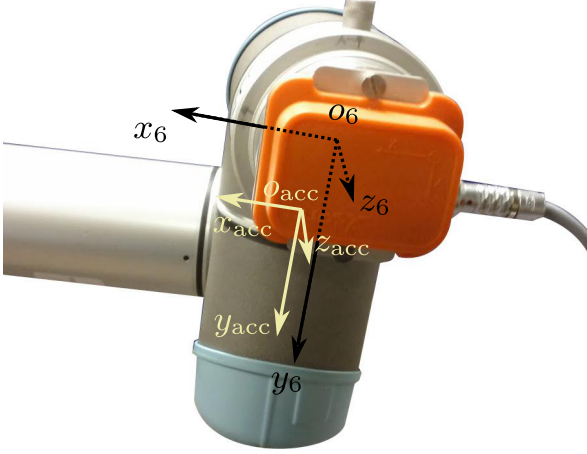


**Figure 5.11.:** Modeled torques of all joints as a result of the applied position and velocity trajectories from Figures 5.1 and 5.2.  $\tau_i$  are the modeled torques as given by the UR5 manipulator, whereas  $\hat{\tau}_i$  are the modeled torques from Model C, for  $i \in \{1, \dots, 6\}$ , in [Nm].



**Figure 5.12.:** Modeling difference  $\epsilon_{\tau,i} = \tau_i - \hat{\tau}_i$  between the modeled torques of all joints, for  $i \in \{1, \dots, 6\}$ , in [Nm].  $\tau_i$  are the modeled torques as given by the UR5 manipulator and  $\hat{\tau}_i$  are the torques from Model C.

## 5.5. Verification of the End Effector Acceleration



**Figure 5.13.:** Accelerometer attached to manipulator. The coordinate frames of the last manipulator link and of the accelerometer are visualized.

The transformation between the coordinate frame that is attached to the last link of the UR5, namely  $o_6x_6y_6z_6$ , and the coordinate frame of the accelerometer, namely  $o_{acc}x_{acc}y_{acc}z_{acc}$ , is visualized in Figure 5.13 and described by

$$\mathbf{T}_{acc}^6 = \begin{bmatrix} 1 & 0 & 0 & 0.01735 \\ 0 & 1 & 0 & 0.00605 \\ 0 & 0 & 1 & 0.0095 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

where the distances are given in meters. This transformation should be taken into account when calculating the end effector acceleration, but was left out in this study for simplification. It is assumed that a qualitative comparison of the accelerometer measurements and the calculated end effector accelerations can be made without including this transformation, as the translation  $\mathbf{o}_{acc}^6 = [17.35, 6.05, 9.5]^T$  mm is small.

The end effector acceleration  $\hat{\mathbf{X}}$  is calculated according to Equation (2.35), where the Euler angles needed for Equations (2.33) and (2.34) are calculated

## 5. UR5 Industrial Manipulator Model

---

from  $\mathbf{T}_6^0$  as described in Section 2.2.1.

The accelerometer measures all accelerations, including the acceleration due to gravity [38]. Therefore gravity has to be subtracted from the measurements at all times. The vector giving the direction of gravity is

$$\mathbf{g}_0 = [0, 0, -9.81]^T \quad (5.10)$$

expressed in the inertial frame  $o_0x_0y_0z_0$  and in  $[m/s^2]$ . The measured accelerations are expressed in the accelerometer coordinate frame  $o_{\text{acc}}x_{\text{acc}}y_{\text{acc}}z_{\text{acc}}$ . By transforming them into the inertial coordinate frame  $o_0x_0y_0z_0$ , the gravity vector  $\mathbf{g}_0$  can be directly subtracted, and the measurements can be compared to the estimated end effector acceleration  $\hat{\dot{\mathbf{X}}}$ , which is expressed in the inertial frame  $o_0x_0y_0z_0$ . The measured accelerations in the accelerometer frame, denoted by  $\mathbf{X}_{\text{acc}}^{\text{acc}}$ , are transformed into the inertial frame and corrected by subtracting the gravity  $\mathbf{g}_0$  from the linear acceleration by calculating

$$\mathbf{X}_{\text{acc}}^0 = \mathbf{R}_6^0 \mathbf{R}_{\text{acc}}^6 \mathbf{X}_{\text{acc}}^{\text{acc}} - \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{0} \end{bmatrix} \quad (5.11)$$

where  $\mathbf{R}_6^0$  is taken from  $\mathbf{T}_6^0$  in Equation (5.3), and  $\mathbf{R}_{\text{acc}}^6 = \mathbf{I} \in \mathbb{R}^{3 \times 3}$  as can be seen in Equation (5.9). The measured acceleration is hereafter denoted by  $\mathbf{X}$  and assumed to be  $\mathbf{X} = \mathbf{X}_{\text{acc}}^0$ .

The comparison of the modeled and measured end effector acceleration over time is shown in Figure 5.14. The error over time is shown in Figure 5.15. Both modeled and measured accelerations were filtered with a lowpass butterworth filter that has a magnitude-squared frequency response

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N_b}} \quad (5.12)$$

where  $N_b$  is the order of the filter and  $\Omega_c$  is the cutoff frequency [19]. The chosen filter order is  $N_b = 8$  and the cutoff frequency is  $\Omega_c = 4.7 \text{ rad/s}$ .

The MAE of the modeled end effector accelerations are

$$\text{MAE}(\hat{\dot{\mathbf{X}}}_i) = \frac{1}{N} \sum_{j=1}^N \left| \dot{X}_{i,j} - \hat{\dot{X}}_{i,j} \right|, \quad i \in \{1, \dots, 6\} \quad (5.13)$$

where  $N$  is the number of measurements that are taken. The MAEs of the modeled accelerations are listed in Table 5.7. It is clear from Figure 5.14 that

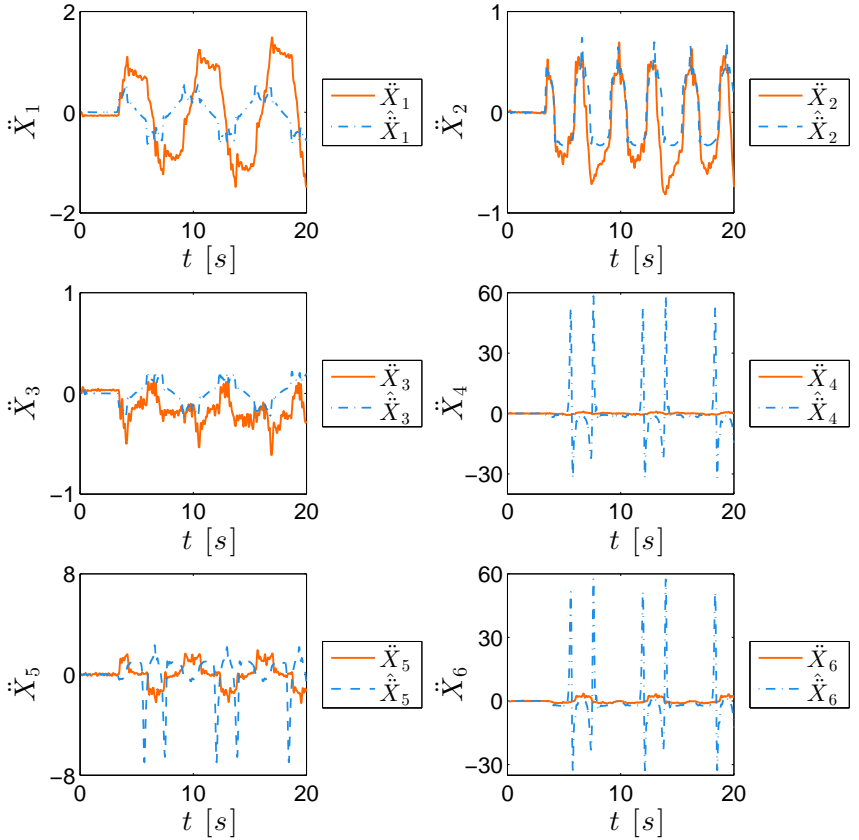


**Table 5.7.:** MAE of the end effector acceleration, where  $\hat{X}_1, \hat{X}_2, \hat{X}_3$  are the modeled linear accelerations in  $x$ -,  $y$ -,  $z$ -direction, and where  $\hat{X}_4, \hat{X}_5, \hat{X}_6$  are the modeled angular accelerations around the  $x$ -,  $y$ -,  $z$ -axes, expressed in the base frame  $o_0x_0y_0z_0$ .

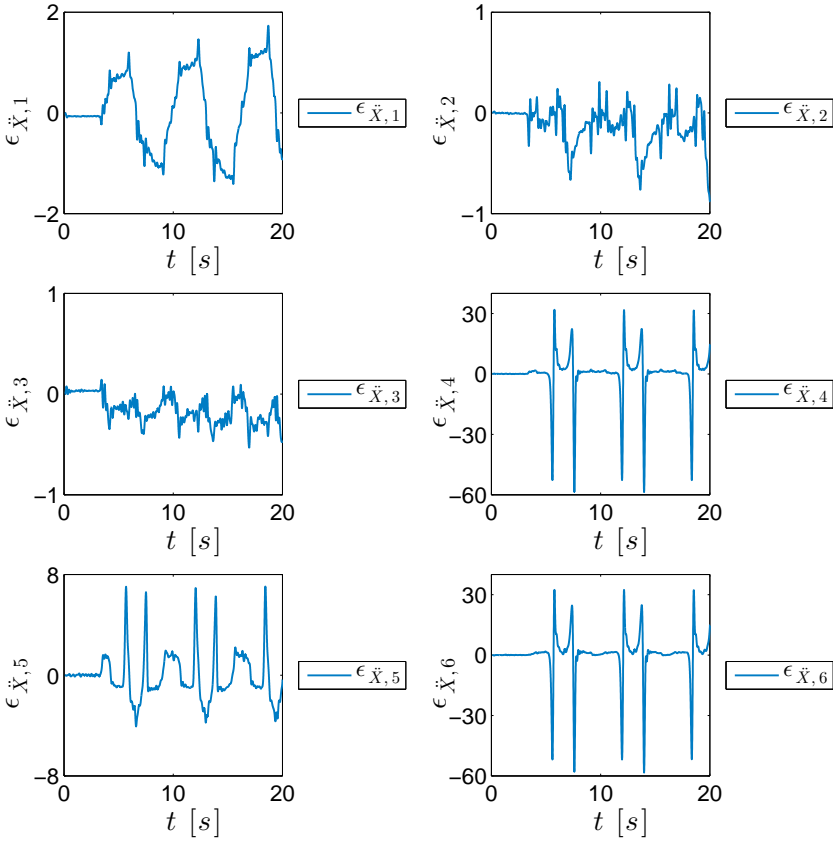
MAE( $\hat{X}_1$ )	MAE( $\hat{X}_2$ )	MAE( $\hat{X}_3$ )	MAE( $\hat{X}_4$ )	MAE( $\hat{X}_5$ )	MAE( $\hat{X}_6$ )
0.6929	0.1810	0.1793	5.1216	1.3968	4.9780

the linear acceleration  $\ddot{X}_2$  is well resembled by the modeled linear acceleration  $\hat{X}_2$ . This is reflected in the low error with  $\text{MAE}(\hat{X}_2) = 0.1810$ . The linear acceleration in  $x$ - and  $z$ -direction over time is less accurate, even though the characteristic periodic movement in the acceleration is resembled by the model. On the contrary, a large error is observed for the angular acceleration. The modeled angular accelerations  $\hat{X}_4, \hat{X}_5$ , and  $\hat{X}_6$  have large peaks at times  $t \in \{5.6, 7.4, 11.95, 13.77, 18.35, 20.15\}$  s. These occur because the manipulator comes close to a singular configuration. A singular configuration exists according to Equation (3.1) at  $\vartheta \in \{k\pi\}$ ,  $k \in \mathbb{Z}$ , which is called a representational singularity. It is found that  $\vartheta = 3.02 \text{ rad} \approx \pi$  at the same time as the peaks in  $\hat{X}_4, \hat{X}_5$ , and  $\hat{X}_6$  appear, i.e., the second Euler angle is close to a representational singularity of the analytical Jacobian. It can be assumed that this singularity also affects the accuracy of the modeled linear acceleration.

The comparison of the modeled and measured accelerations suggests that the accelerometer is well embedded into the system. More work is needed prior to the use of the accelerometer as a means for force estimation, such as including  $T_{\text{acc}}^6$  into the end effector acceleration model and the handling of singularities. The further use of the accelerometer is therefore beyond the scope of this study and is suggested for future work.



**Figure 5.14.:** Modeled and measured acceleration at the end effector.  $\ddot{X}_1, \ddot{X}_2, \ddot{X}_3$  are the measured and  $\hat{X}_1, \hat{X}_2, \hat{X}_3$  are the modeled linear accelerations in  $x$ -,  $y$ -,  $z$ -direction, respectively.  $\ddot{X}_4, \ddot{X}_5, \ddot{X}_6$  are the measured and  $\hat{X}_4, \hat{X}_5, \hat{X}_6$  are the modeled angular accelerations around the  $x$ -,  $y$ -,  $z$ -axes, respectively. All accelerations are expressed in the base frame  ${}_{00}x_0y_0z_0$ . The units are  $[m/s^2]$  for the linear acceleration and  $[rad/s^2]$  for the angular acceleration.



**Figure 5.15.:** Modeling error of the acceleration at the end effector.  $\epsilon_{\ddot{X},i} = \ddot{X}_i - \hat{\ddot{X}}_i$ , for  $i \in \{1, 2, 3\}$ , are the modeling errors for the linear accelerations in  $x$ -,  $y$ -,  $z$ -direction, respectively.  $\epsilon_{\ddot{X},i} = \ddot{X}_i - \hat{\ddot{X}}_i$ , for  $i \in \{4, 5, 6\}$ , are the modeling errors for the angular accelerations around the  $x$ -,  $y$ -,  $z$ -axes, respectively. All errors are expressed in the base frame  $o_0x_0y_0z_0$ . The units are  $[m/s^2]$  for  $\epsilon_{\ddot{X},1}$ ,  $\epsilon_{\ddot{X},2}$ ,  $\epsilon_{\ddot{X},3}$  and  $[rad/s^2]$  for  $\epsilon_{\ddot{X},4}$ ,  $\epsilon_{\ddot{X},5}$ ,  $\epsilon_{\ddot{X},6}$ .

## 5.6. Summary and Conclusion

### 5.6.1. Summary

The dynamic model derived by the author in [21] was verified in a closed loop and showed large torque modeling errors. A closer look at the dynamic equations suggested that the model equations were not correct and that a new model might improve the modeling performance. Three new models were derived with different link modeling techniques. All three models showed better performance in modeling the torques when compared to the model in [21]. The two models that were derived using knowledge about the mass center points in the links showed the highest accuracy. Their performance was surprisingly similar, which might be due to the low weight of the UR5. Nevertheless, the model with the most accurate modeling method of the inertia tensors was chosen for the remainder of the thesis.

Finally, acceleration measurements at the end effector were compared to the modeled end effector acceleration. The comparison suggests that the accelerometer is correctly embedded into the experimental setup, but that more work has to be done before it can be used as a means for force estimation. The accuracy of the modeled acceleration was affected when the UR5 moved close to a representational singularity, which emphasizes the need for a solution for the cases where the manipulator comes close to a singular configuration.

### 5.6.2. Conclusion

In order to obtain an accurate inertia tensor model it is more important to model a realistic weight distribution of a link instead of modeling its geometric shape. The knowledge of the mass center point of each link is therefore a very important aid in deriving an accurate dynamic model. They should therefore always be taken into account in the modeling process when they are available.

Mass center points that are not at the centroid of a link can be reconstructed by using multiple geometric bodies with different mass to resemble the link. This method does not improve the accuracy of a dynamic model for a lightweight manipulator, but is recommended for manipulators with heavier links.

Singularities play a major role in the calculation of the end effector acceleration. They should therefore be calculated for each manipulator and avoided when the trajectories are planned.

## 6. Force Estimation on the UR5

Force estimation is implemented in this chapter, based on torque estimates within the joints. The torques in turn are estimated based on the dynamic equations for the UR5 and the current-torque relationship, which is established in Section 6.1. It is shown that friction modeling plays an important role in finding a well defined relationship between currents and torques. Experimental results are presented in Section 6.2, where the force estimates coming from the UR5 and force estimates based on currents are compared.

### 6.1. Current Torque Relationship

In order to estimate external forces that are acting on the manipulator according to Equation (3.7), it is important to find a current-torque relationship such as, for example, in Equation (3.6). At first it is necessary to find out if the modeled torque is proportional to the measured current. Initially, friction is neglected and a relationship between estimated torques based on the dynamic equations  $\hat{\boldsymbol{\tau}}_\mu$  and the currents  $\mathbf{i}_a$  is sought. An experiment is performed where the currents are recorded for a random trajectory, without external forces acting on the UR5. The currents are then illustrated in relation to the estimated torques in Figure 6.1. It is clear from the visualization that a linear relationship between  $\hat{\boldsymbol{\tau}}_\mu$  and  $\mathbf{i}_a$  can not be established.

It is then verified whether friction modeling helps to find a current-torque relationship. The torque friction in the joints is therefore modeled according to Equations (3.12), (3.13) and (3.14). The coulomb and viscous friction coefficients are adjusted manually with a heuristic approach, where

$$\min_{\boldsymbol{\nu}_c, \boldsymbol{\nu}_v, \tilde{\mathbf{k}}_\tau} \mathcal{J}(\boldsymbol{\nu}_c, \boldsymbol{\nu}_v, \tilde{\mathbf{k}}_\tau) = \sum_{j=1}^N (\boldsymbol{\tau}_\mu(t_j) + \boldsymbol{\tau}_{fr}(t_j, \boldsymbol{\nu}_c, \boldsymbol{\nu}_v) - \tilde{\mathbf{k}}_\tau^T \mathbf{i}_a(t_j)) \quad (6.1)$$

s.t.  $\boldsymbol{\tau}_\mu(t_j)$  is the solution of Equation (3.8) and

$$\boldsymbol{\tau}_{fr}(t_j, \boldsymbol{\nu}_c, \boldsymbol{\nu}_v) \text{ is the solution of Equations (3.12) – (3.14)}$$

and where  $\tilde{\mathbf{k}}_\tau$  is a preliminary torque constant vector which will be optimized

later. The chosen friction coefficients are

$$\boldsymbol{\nu}_c = \boldsymbol{\nu}_v = [7.2, 5.5, 5.7, 0.7, 0.7, 0.65]^T \quad (6.2)$$

with the preliminary torque constant vector

$$\tilde{\mathbf{k}}_\tau = [9.1, 8.3, 8.3, 4, 4, 4]^T \quad (6.3)$$

A dead zone is implemented to avoid immediate switching of the torque friction when  $\dot{q}_i$  crosses zero, which can happen frequently in a short period of time and cause unwanted oscillations in the simulation, as it is run with fixed time step. Equation (3.15) is therefore changed to

$$\tau_{cou,i} = \begin{cases} -\nu_{c,i}, & \dot{q}_i < -\delta \\ 0, & |\dot{q}_i| \leq \delta \\ \nu_{c,i}, & \dot{q}_i > \delta \end{cases} \quad (6.4)$$

with  $\delta = 0.02$ .

Again, currents  $\mathbf{i}_a$  are plotted against the estimated torques  $\hat{\boldsymbol{\tau}}$ , but this time including a friction model, such that  $\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_\mu + \hat{\boldsymbol{\tau}}_{fr}$ . The plots are given in Figure 6.2. A constant relationship between modeled torques and measured currents is clearly recognizable. A linear model with offset is used to fit the data, which yields

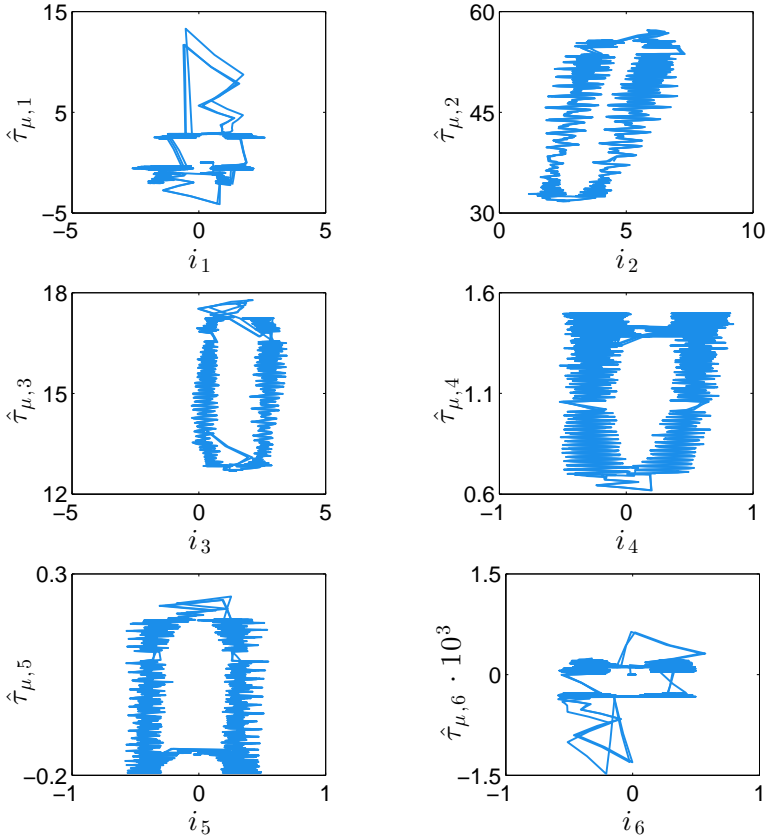
$$\begin{aligned} \hat{\tau}_1 &= 7.5i_1 - 0.61 \\ \hat{\tau}_2 &= 6.8i_2 + 16 \\ \hat{\tau}_3 &= 7.1i_3 + 4.5 \\ \hat{\tau}_4 &= 2.8i_4 + 0.84 \\ \hat{\tau}_5 &= 3.2i_5 + 0.064 \\ \hat{\tau}_6 &= 3.4i_6 + 0.099 \end{aligned} \quad (6.5)$$

where  $\mathbf{i}_a = [i_1, i_2, i_3, i_4, i_5, i_6]^T$ . Equations (6.5) can be rewritten into

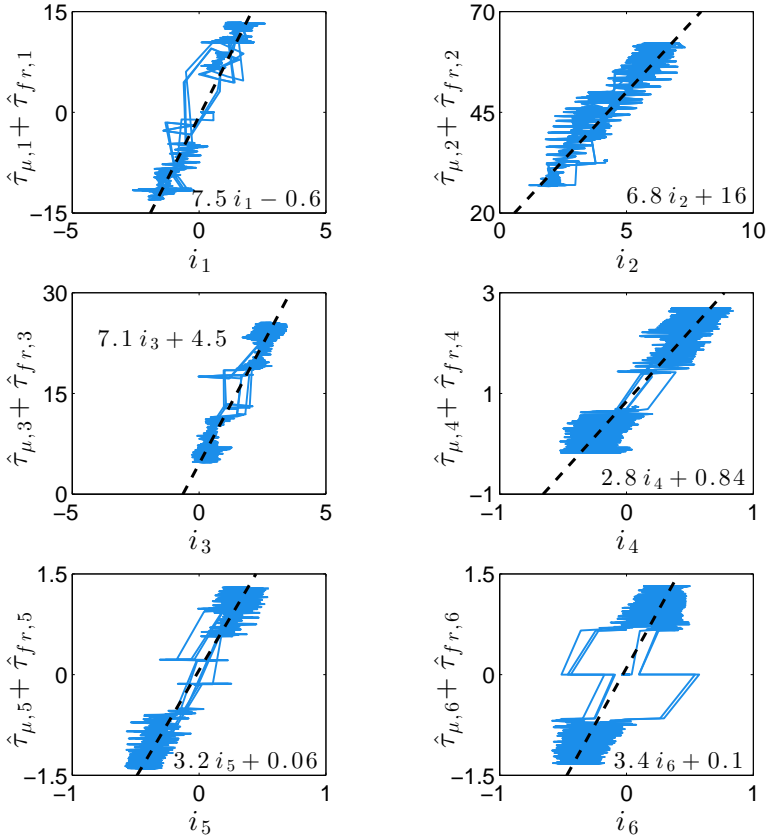
$$\begin{aligned} \hat{\boldsymbol{\tau}} &= \mathbf{k}_\tau^T \mathbf{i}_a + \mathbf{b} \\ \mathbf{k}_\tau &= [7.5, 6.8, 7.1, 2.8, 3.2, 3.4]^T \\ \mathbf{b} &= [-0.61, 16, 4.5, 0.84, 0.064, 0.099]^T \end{aligned} \quad (6.6)$$

where  $\mathbf{k}_\tau$  is the optimized torque constant vector.

This relationship is used for force estimation in the next section.



**Figure 6.1.:** Current-torque relationship, without friction. The measured currents  $i_a = [i_1, i_2, i_3, i_4, i_5, i_6]^T$ , in [A], are plotted against the estimated torques  $\hat{\tau}_\mu$ , in [Nm], in order to find a quantifiable function that describes the relationship.



**Figure 6.2.:** Current torque relationship, with friction. The measured currents  $i_a = [i_1, i_2, i_3, i_4, i_5, i_6]^T$ , in [A], are plotted against the estimated torques  $\hat{\tau}_{\mu} + \hat{\tau}_{fr}$ , in [Nm], in order to find a quantifiable function that describes the relationship. The relationship can be approximated by linear functions.



## 6.2. Force Estimation at the End Effector

A linear current-torque relationship was found in the previous section, where the torque model was based on torques due to dynamics and friction. In the case that an external force is acting on the manipulator, this force will be reflected in additional torques in the joints, which in turn will induce motor currents. It is assumed that the constant function describing the current-torque relationship also holds for torques that are induced by external forces. This means that Equation (3.6) can be used. It is modified to

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_{\mu} + \hat{\boldsymbol{\tau}}_{ext} + \hat{\boldsymbol{\tau}}_{fr} = \mathbf{k}_{\tau}^T \mathbf{i}_a + \mathbf{b} \quad (6.7)$$

in order to match Equation (6.6). The external force can then be estimated according to Equation (3.3). In order to compare the performance of force estimation with and without friction modeling, both estimates are considered, where

$$\hat{\mathbf{F}} = (\mathbf{J}_g^T(\mathbf{q}))^{-1} \hat{\boldsymbol{\tau}}_{ext} = (\mathbf{J}_g^T(\mathbf{q}))^{-1} (\mathbf{k}_{\tau}^T \mathbf{i}_a + \mathbf{b} - \hat{\boldsymbol{\tau}}_{\mu}) \quad (6.8)$$

is the force estimate without considering friction and where

$$\hat{\mathbf{F}}^* = (\mathbf{J}_g^T(\mathbf{q}))^{-1} \hat{\boldsymbol{\tau}}_{ext} = (\mathbf{J}_g^T(\mathbf{q}))^{-1} (\mathbf{k}_{\tau}^T \mathbf{i}_a + \mathbf{b} - \hat{\boldsymbol{\tau}}_{\mu} - \hat{\boldsymbol{\tau}}_{fr}) \quad (6.9)$$

is the force estimate with modeled friction. Furthermore, a third force estimate is available from the UR5 which is used for comparison. It is denoted by  $\hat{\mathbf{F}}^u$  from here on.

In order to evaluate all three force estimates, the ATI force sensor is used. The setup is illustrated in Figure 6.3, together with the coordinate frames that are assigned to link 6 and to the force sensor. The transformation between the coordinate frames, namely  $o_6x_6y_6z_6$  and  $o_{ft}x_{ft}y_{ft}z_{ft}$ , is given by

$$\mathbf{T}_{ft}^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.0286 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.10)$$

where the distances are given in meters. As can be seen from the transformation matrix, the force sensor was attached such that the rotation matrix  $\mathbf{R}_{ft}^6 = \mathbf{I} \in \mathbb{R}^{3 \times 3}$ . There is only a small distance between the origins of the two coordinate frames, which was neglected in the calculation of the Jacobian  $\mathbf{J}_g^T$ . The force measurements are given in the coordinate frame of the force sensor, namely  $o_{ft}x_{ft}y_{ft}z_{ft}$ , while the estimated forces  $\hat{\mathbf{F}}$ ,  $\hat{\mathbf{F}}^*$  and  $\hat{\mathbf{F}}^u$  are expressed

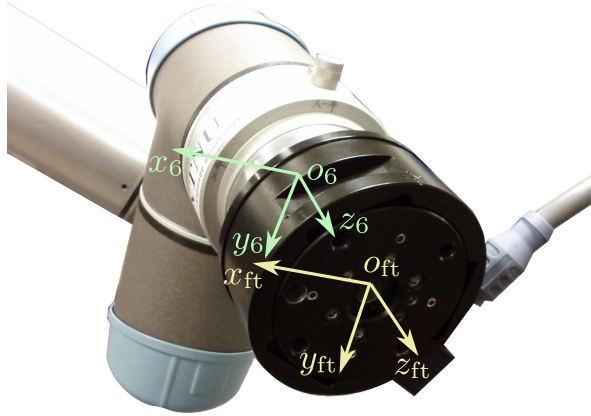
## 6. Force Estimation on the UR5

---

in the inertial frame  $o_0x_0y_0z_0$ . It is therefore necessary to transform the force measurements  $\mathbf{F}_{ft}^{ft}$  that are expressed in the force sensor frame, into the inertial frame by using the calculation

$$\mathbf{F}_{ft}^0 = \mathbf{R}_6^0 \mathbf{R}_{ft}^6 \mathbf{F}_{ft}^{ft} \quad (6.11)$$

where  $\mathbf{R}_6^0$  is taken from  $\mathbf{T}_6^0$  in Equation (5.3), and  $\mathbf{R}_{ft}^6 = \mathbf{I} \in \mathbb{R}^{3 \times 3}$  as can be seen in Equation (6.10). The measured force will be denoted by  $\mathbf{F}$  and assumed to be  $\mathbf{F} = \mathbf{F}_{ft}^0$ .



**Figure 6.3.:** Force sensor attached to manipulator. The coordinate frames of the last manipulator link and of the force sensor are visualized.

The current measurements are noisy, which influences the force estimates  $\hat{\mathbf{F}}$  and  $\hat{\mathbf{F}}^*$ . A lowpass butterworth filter is used to reduce the noise. It has a magnitude-squared frequency response as described in Equation (5.12). The chosen filter order is  $N_b = 5$  and the cutoff frequency is  $\Omega_c = 1.454 \text{ rad/s}$ .

The results of the force estimation from the UR5 and from the model that does not include friction are visualized in the Figures 6.4 and 6.5. The results of the force estimation from the model that includes friction are visualized in the Figures 6.6 and 6.7. The errors of all three force estimates are visualized in Figures 6.8 and 6.9. The MAE of the modeled end effector forces and torques

**Table 6.1.:** Comparison of MAE of force estimates  $\hat{F}_i^u$ ,  $\hat{F}_i$ ,  $\hat{F}_i^*$  and torque estimates  $\hat{n}_i^u$ ,  $\hat{n}_i$ ,  $\hat{n}_i^*$ , for  $i \in \{x, y, z\}$ . Smallest errors are colored in green and largest errors are colored in red.

	MAE( $\hat{F}^u$ )	MAE( $\hat{F}$ )	MAE( $\hat{F}^*$ )
$F_x$	7.8240	37.8064	6.1571
$F_y$	16.9189	52.3023	8.6588
$F_z$	2.9237	10.0739	5.6363
$n_x$	2.6154	3.1774	0.9955
$n_y$	3.4000	3.8464	1.1750
$n_z$	1.9967	1.4735	1.2271

are

$$\text{MAE}(\hat{F}_i) = \frac{1}{N} \sum_{j=1}^N |F_{i,j} - \hat{F}_{i,j}|, \quad i \in \{x, y, z\} \quad (6.12)$$

$$\text{MAE}(\hat{n}_i) = \frac{1}{N} \sum_{j=1}^N |n_{i,j} - \hat{n}_{i,j}|, \quad i \in \{x, y, z\}$$

respectively, where  $N$  is the number of measurements that are taken. The maximum absolute errors are calculated by finding the maximum value among all absolute errors

$$\Delta(\hat{\mathbf{F}})(t_k) = |\mathbf{F}(t_k) - \hat{\mathbf{F}}(t_k)| \quad (6.13)$$

which are calculated for all time instances  $t_k$ ,  $k \in \{1, \dots, N\}$ . The MAEs and maximum absolute errors of  $\hat{\mathbf{F}}^*$  and  $\hat{\mathbf{F}}^u$  are calculated in the same way as for  $\hat{\mathbf{F}}$ . The MAEs of the force estimates are listed in Table 6.1 and the maximum absolute errors are listed in Table 6.2.

It is clear from the plots and calculated errors that the force estimation  $\hat{\mathbf{F}}$  that is based on a torque model without friction modeling has the largest error. The exerted force in  $x$ - and  $y$ -direction is barely noticeable in the force estimation  $\hat{F}_x$  and  $\hat{F}_y$ , as the error is by far larger than the forces that are acting in those directions. The force exerted in  $z$ -direction is partly reflected by  $\hat{F}_z$ , but the error  $\epsilon_{F_z}$  is still very large with a MAE( $\hat{F}_z$ ) = 10.0739.

**Table 6.2.:** Comparison of maximum absolute errors of force estimates  $\hat{F}_i^u$ ,  $\hat{F}_i$ ,  $\hat{F}_i^*$  and torque estimates  $\hat{n}_i^u$ ,  $\hat{n}_i$ ,  $\hat{n}_i^*$ , for  $i \in \{x, y, z\}$ . Smallest errors are colored in green and largest errors are colored in red.

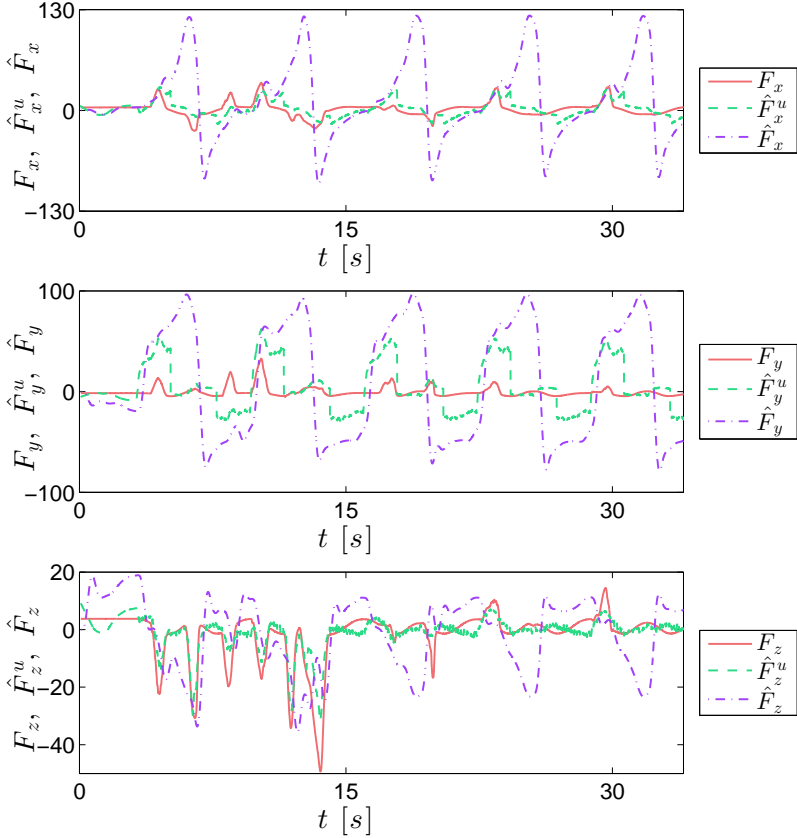
	$\max  F - \hat{F}^u $	$\max  F - \hat{F} $	$\max  F - \hat{F}^* $
$F_x$	25.0072	139.499	32.0353
$F_y$	48.8422	98.0112	27.7541
$F_z$	21.1399	35.3182	28.1711
$n_x$	8.7412	7.1011	6.2946
$n_y$	9.9929	11.7200	5.0910
$n_z$	5.8017	6.3418	2.1928

The force estimation  $\hat{F}^u$  coming from the UR5 reflects the exerted forces in  $x$ - and  $z$ -direction well. It has the lowest MAE for  $\hat{F}_z$  and the lowest maximum absolute error for  $\hat{F}_x$  and  $\hat{F}_z$  when compared to  $\hat{F}^u$  and  $\hat{F}^*$ . The errors in  $y$ -direction are large and this method takes the middle position in the ranking of accuracy for all three estimates.

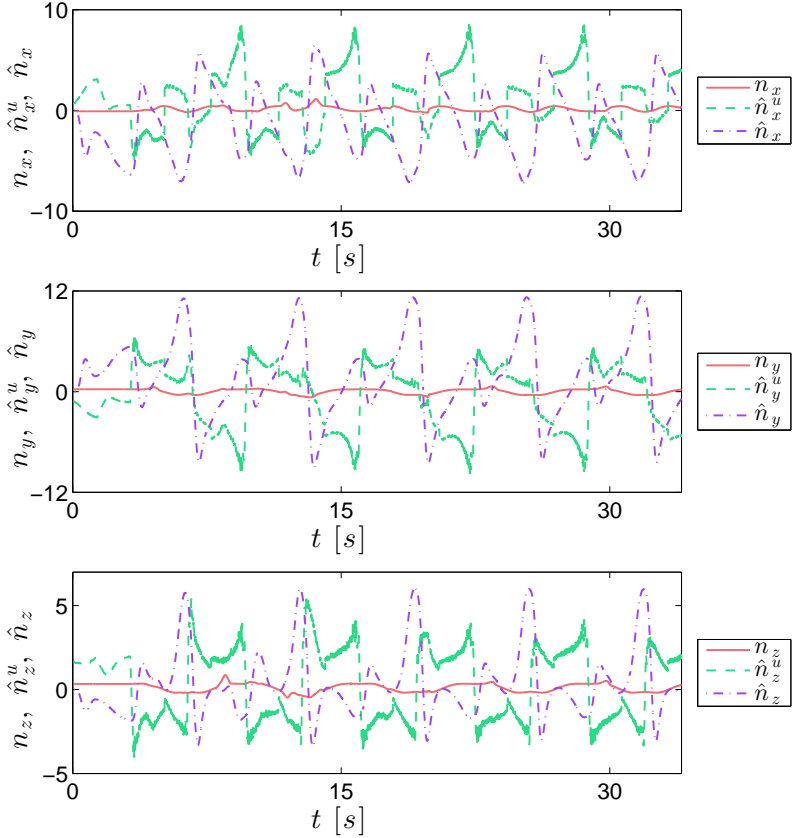
The force estimation  $\hat{F}^*$  that is based on a torque model which includes a friction model has the highest accuracy. The forces acting on the manipulator in  $x$ - and  $z$ -direction are reflected in the force estimates  $\hat{F}_x^*$  and  $\hat{F}_z^*$ . The force estimate  $\hat{F}_y^*$  in  $y$ -direction shows the lowest error when compared to the other two force estimates  $\hat{F}_y$  and  $\hat{F}_y^u$ .

None of the force estimation methods reflect the torques that are acting on the end effector. While almost no torques are measured from the F/T sensor, all estimation methods predict end effector torques, where the MAEs as well as the maximum absolute errors are higher for  $\hat{n}_i^u$  and  $\hat{n}_i$  than for  $\hat{n}_i^*$ , for  $i \in \{x, y, z\}$ .

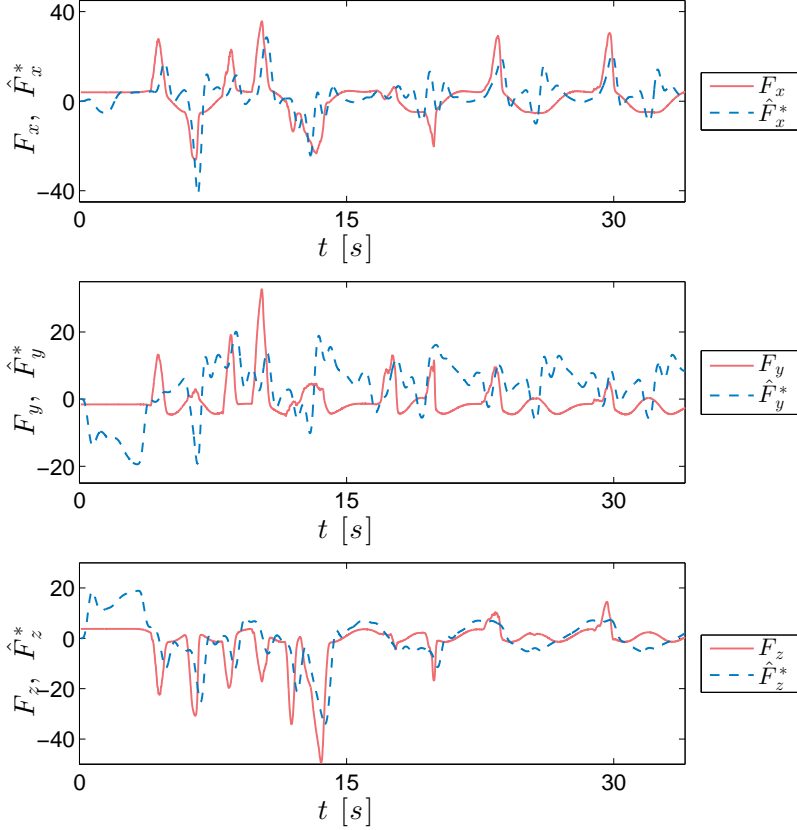
The comparison of the modeled and measured forces gives rise to the conclusion that the torque estimates are unreliable. In comparison to the force estimates in both the  $x$ - and  $y$ - directions, the force estimates in  $z$ -direction are the most accurate for all of the force estimation methods. Among the methods that were implemented in this thesis, the model that includes a friction model gives the most accurate force estimates in  $z$ -direction.  $\hat{F}_z^*$  can be used for collision detection, as forces  $\geq 10 N$  are detected reliably.



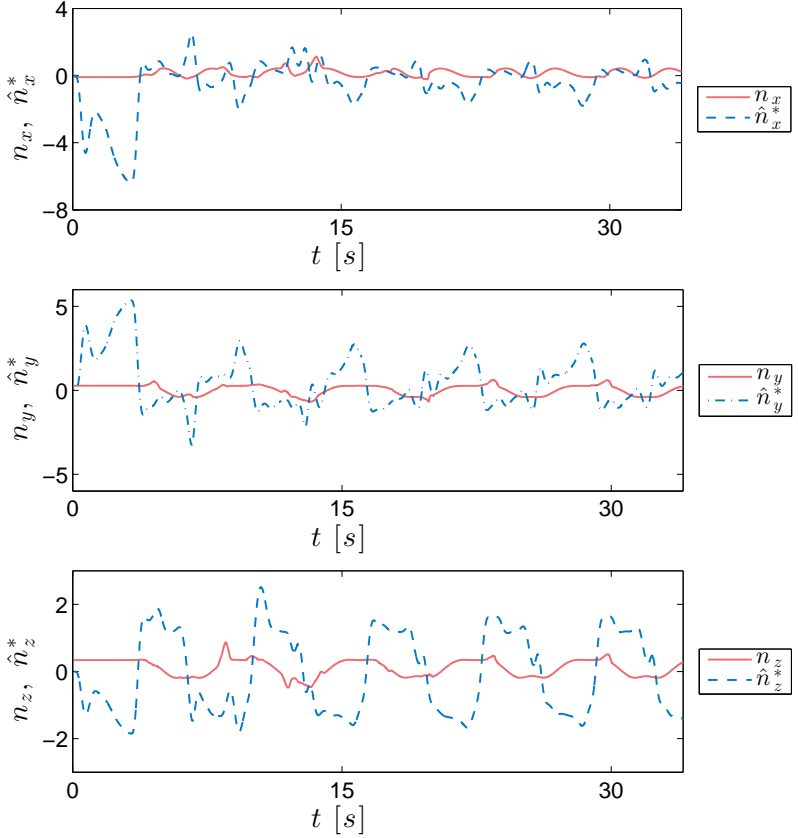
**Figure 6.4.:** Comparison of estimates for the external forces acting on the end effector of the UR5 manipulator. The force estimates coming from the UR5 manipulator, denoted by  $\hat{\mathbf{F}}^u = [\hat{F}_x^u, \hat{F}_y^u, \hat{F}_z^u, \hat{n}_x^u, \hat{n}_y^u, \hat{n}_z^u]^T$ , and from the torque model without considering friction, denoted by  $\hat{\mathbf{F}} = [\hat{F}_x, \hat{F}_y, \hat{F}_z, \hat{n}_x, \hat{n}_y, \hat{n}_z]^T$ , are compared to the force measurements from the force sensor, denoted by  $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z]^T$ . The units for  $F$  are [N].



**Figure 6.5.:** Comparison of estimates for the external forces acting on the end effector of the UR5 manipulator. The force estimates coming from the UR5 manipulator, denoted by  $\hat{\mathbf{F}}^u = [\hat{F}_x^u, \hat{F}_y^u, \hat{F}_z^u, \hat{n}_x^u, \hat{n}_y^u, \hat{n}_z^u]^T$ , and from the torque model without considering friction, denoted by  $\hat{\mathbf{F}} = [\hat{F}_x, \hat{F}_y, \hat{F}_z, \hat{n}_x, \hat{n}_y, \hat{n}_z]^T$ , are compared to the force measurements from the force sensor, denoted by  $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z]^T$ . The units for  $n$  are  $[Nm]$ .

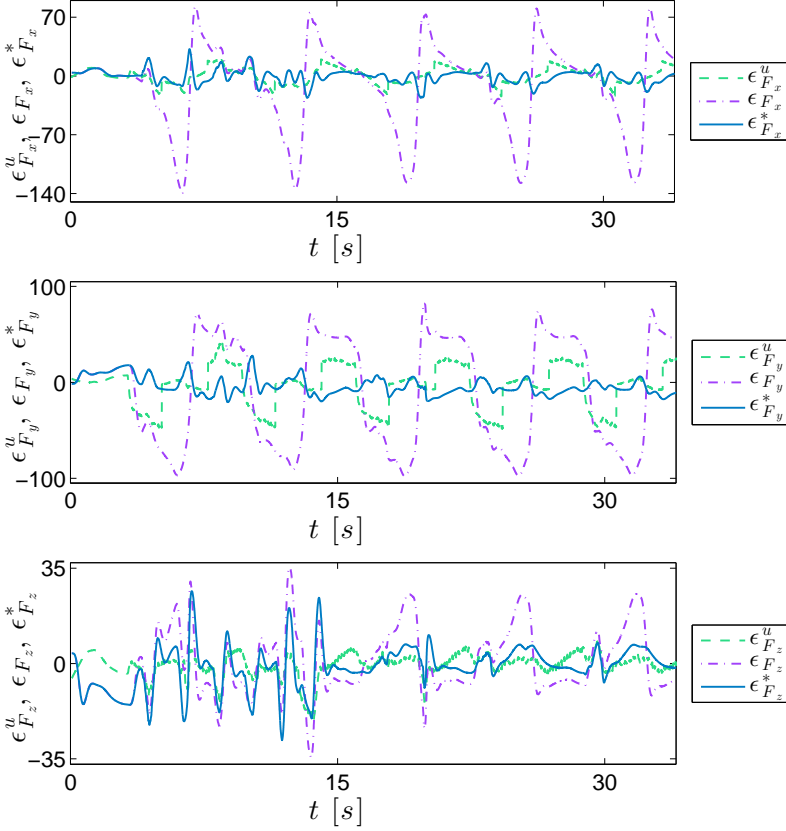


**Figure 6.6.:** Comparison of estimates for the external forces acting on the end effector of the UR5 manipulator. The force estimates from the torque model that considers friction, denoted by  $\hat{\mathbf{F}}^* = [\hat{F}_x^*, \hat{F}_y^*, \hat{F}_z^*, \hat{n}_x^*, \hat{n}_y^*, \hat{n}_z^*]^T$ , are compared to the force measurements from the force sensor, denoted by  $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z]^T$ . The units for  $\mathbf{F}$  are  $[N]$ .

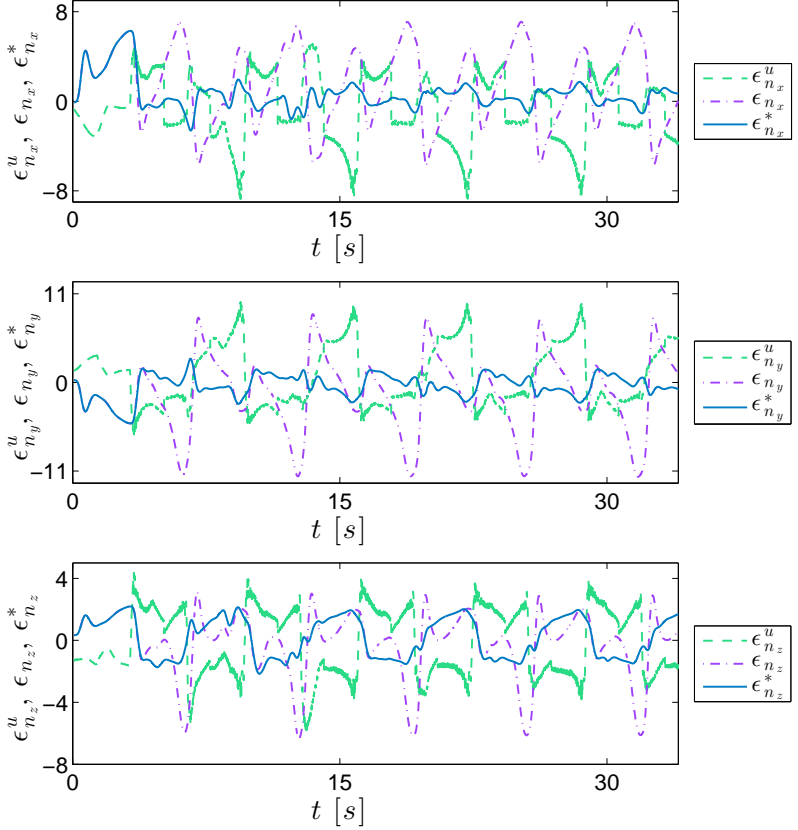


**Figure 6.7.:** Comparison of estimates for the external forces acting on the end effector of the UR5 manipulator. The force estimates from the torque model that considers friction, denoted by  $\hat{\mathbf{F}}^* = [\hat{F}_x^*, \hat{F}_y^*, \hat{F}_z^*, \hat{n}_x^*, \hat{n}_y^*, \hat{n}_z^*]^T$ , are compared to the force measurements from the force sensor, denoted by  $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z]^T$ . The units for  $n$  are [Nm].





**Figure 6.8.:** Estimation errors of the force estimates  $\hat{F}_i^u$ ,  $\hat{F}_i$ ,  $\hat{F}_i^*$  for the external forces acting on the end effector of the UR5 manipulator, where  $\epsilon_{F_i}^u = F_i - \hat{F}_i^u$ ,  $\epsilon_{F_i} = F_i - \hat{F}_i$ ,  $\epsilon_{F_i}^* = F_i - \hat{F}_i^*$ , for  $i \in \{x, y, z\}$ . The units for  $\epsilon_F$  are [N].



**Figure 6.9.:** Estimation errors of the torque estimates  $\hat{n}_i^u$ ,  $\hat{n}_i$ ,  $\hat{n}_i^*$  for the external forces acting on the end effector of the UR5 manipulator, where  $\epsilon_{n_i}^u = n_i - \hat{n}_i^u$ ,  $\epsilon_{n_i} = n_i - \hat{n}_i$ ,  $\epsilon_{n_i}^* = n_i - \hat{n}_i^*$ , for  $i \in \{x, y, z\}$ . The units for  $\epsilon_n$  are  $[Nm]$ .

## 6.3. Summary and Conclusion

### 6.3.1. Summary

By using a torque model based on the equations of dynamics for the UR5 manipulator, together with a friction model for the friction in the joints, a relationship between the currents measurements and the modeled torques is found. The relationship is described by a constant function with offset for each joint. The parameters for the friction torque model were adjusted manually. Data fitting was then used to map the currents to the modeled torques. Force estimation was then performed based on this mapping, both on a model that considers friction and on a model that does not. The difference between the estimated torques, based on the measured currents, and the expected torques in the joints indicates the presence and magnitude of the contact forces. Those two estimates were compared with the force estimates coming from the UR5 manipulator and with force measurements from the ATI force sensor. The comparison shows that none of the force estimates comes close to the accuracy of a force sensor. The torque estimates are unreliable for all force estimation methods. From the methods that were implemented in this thesis, the force estimate based on a torque model with friction modeling has the highest accuracy, especially in  $z$ -direction. It can be therefore used to detect contact forces.

### 6.3.2. Conclusion

A simple force estimation method based on current measurements and a torque model with friction modeling for the joints of a manipulator can be used to detect contact forces. The estimation is most reliable for linear forces and least reliable for torques acting on the end effector. One possible reason for the high force estimation errors is the suboptimal choice of the friction model parameters, as those were adjusted manually. This could be improved by running an optimization algorithm to find the friction model parameters together with the parameters for the linear function that describe the current-torque relationship.

Furthermore, the force estimation method was only applied on the same trajectory that was used to establish the current-torque relationship. The performance of the estimation technique can change when the manipulator follows a new trajectory, as friction can be position dependent [25]. An adaptive observer could be used to continuously update the parameters of the friction model and of the current-torque relationship, such that contact forces can be detected for arbitrary positions of the manipulator.

## 6. Force Estimation on the UR5

---

Singular configurations have to be avoided in order to estimate the forces successfully. The task of finding the singularities for the UR5 is therefore important, as well as finding a solution of how to avoid the vicinity of singular configurations.

## 7. Conclusion and Outlook

A summary and conclusion of this thesis is given in Section 7.1. Section 7.2 gives recommendations for future work.

### 7.1. Conclusion

The goal of this project was to implement force estimation in order to detect forces at the end effector of the robot. This includes setting up a test stand with the industrial manipulator UR5, an ATI F/T sensor, and a Xsens MTi accelerometer. The first step was to set up the equipment and to implement software for each device. Python is used to communicate via sockets to the UR5 controller and to the force sensor. The control of the robot and obtaining measurements are based upon requests. As all requests are blocking calls, threads are used to parallelize the program. The accelerometer program is based on provided C++ libraries. A bash script is used to start all modules simultaneously and to terminate all the concurrent programs. The data can then be viewed instantly by using the Python library matplotlib or forwarded to Matlab, where it can be used for simulations and further data processing. In this environment, Python has been an excellent tool to connect devices and prototype experiments efficiently.

The second step towards force estimation was to obtain and verify a correct model of the UR5 manipulator. A model of the manipulator was derived by the author in [21], which had yet to be verified. This could only be done through a closed loop, as no access to the low level controller was given. It was shown that the control structure can be assumed to be a feedback linearization with an overlying PD controller, and PD control parameters could be estimated. A comparison of the modeled torque with the torques coming from the UR5 manipulator showed that the model derived in [21] was inaccurate.

Based on this evidence, new dynamic models for the UR5 were derived by using the Euler-Lagrange method and different modeling techniques for the inertia tensors. Special attention in the modeling is paid to the non-uniform mass distribution in the links of the manipulator. Three models were derived and compared: Model A where the modeling of the link shapes was prioritized

over the position of the mass center point, Model B where the mass center point for each link was correct, but where the link shapes were modeled poorly, and Model C, where both link shape as well as the non-uniform mass distribution were accurately modeled. All three models showed better performance in modeling the torques when compared to the model in [21]. A comparison of the performance of the three models shows that the torque modeling error is much higher for Model A than for Model B and C. The performance of the latter two models is similar, even though it was expected that Model C would be more accurate than Model B. A possible explanation for the similar performance is the low weight of the UR5, where the dynamic equations are not affected by changes in the inertia tensors as much as in heavier industrial manipulators. The shape of the lightweight links does not affect the inertia tensors enough to make a difference in the accuracy of the dynamic model. The position of the mass center points on the other hand has a high impact on the performance of the model. It has to be noted that the performance comparisons are made with respect to torque estimates from the UR5 and not with respect to torque measurements. The torque modeling error can therefore not be used as an absolute performance measure, but rather as a guidance. As Model C models both link shapes and mass distribution most accurately, it was chosen as the dynamic model for the remainder of the thesis. It can be concluded that in order to obtain an accurate inertia tensor model it is more important to model a realistic weight distribution of a link instead of modeling its geometrical shape. The knowledge of the mass center point of each link should always be incorporated in the modeling process when their position is known. Mass center points that are not at the centroid of a link can be reconstructed by using multiple geometric bodies with different mass to resemble the link.

After the derivation of the manipulator dynamic equations, the modeled acceleration at the end effector was compared to the accelerometer measurements. While the modeled linear accelerations in the direction of the  $y$ - and  $z$ -axis resemble the measurements, the angular accelerations are distorted as the manipulator moves close to a singular configuration. From the experimental results it can be concluded that the accelerometer is correctly embedded into the experimental setup. The accelerometer measurements are more reliable than the modeled acceleration when the manipulator is close to a singular configuration. Prior to further use of the accelerometer as a means for force estimation, all singular configurations of the UR5 must be identified and a solution found for the cases where the manipulator comes close to those configurations.

Finally, force estimation was performed based on current measurements. The difference between the estimated torques, based on the measured currents, and

the expected torques in the joints indicates the presence and magnitude of the contact forces. Only by modeling the torques based on both dynamics and friction could the constant current-torque relationship be found. Three force estimates were computed and compared with the measurements of a force sensor: A force estimate based on a torque model that neglects the torque friction, a force estimate based on a torque model with friction modeling, and a force estimate that was available from the UR5 manipulator. The comparison shows that the force estimate based on the torque model that includes a friction model reaches the highest accuracy, which emphasizes the importance of friction modeling for force estimation. The force estimate in  $z$ -direction for this method has the highest accuracy, where forces  $\geq 10\text{ N}$  are detected. It can therefore be used for collision detection.

A groundwork for research on force estimation with accelerometers was laid in this thesis. The physical experiment including all devices was set up and mathematically modeled. Deriving an accurate dynamic model for the UR5, the end effector acceleration, the torque friction and the current-torque relationship is crucial for successful force estimation, as all modeling errors result in an error in the force estimation. Preliminary force estimation results were achieved in this work, despite a closed control architecture. By using additional measurements from the accelerometer, both the dynamic and friction model can be tested more thoroughly and improved, leading eventually to more accurate force estimates.

Recommendations for how to improve the obtained results and for future work are given in the next section.

## 7.2. Outlook

A major restriction in the research and development of force estimation in this study was the closed control architecture of the UR5 manipulator. It is therefore crucial to acquire the C-API from Universal Robot and implement real-time control. This enables implementation of real-time trajectory tracking and force control based on the previously implemented force estimation.

More studies should be performed with the inertia tensor modeling technique for manipulator links with non-uniform mass distribution. More sophisticated inertia tensor modeling methods might improve the model accuracy, especially for heavy industrial manipulators, because the inertia tensors increasingly impact the dynamic equations as the link weight increases.

The accuracy of the force estimation method can easily be improved by imple-

menting an optimization algorithm to estimate the friction model parameters together with the mapping between the measured currents and the modeled torques. The algorithm can be extended even further by the use of adaptive observers to identify those parameters, such that the force estimation method becomes robust for all positions of the manipulator.

Studies on observability of external forces should be performed in different states of the robot, such as static, semi-static and dynamic states. Friction is most challenging to model for a static state of the manipulator, such that forces might not be observable. Experiments in semi-static states can help to determine the minimum velocity where friction estimation becomes feasible.

Finding the singularities of the UR5 is fundamental both for the calculation of the end effector acceleration and for the force estimation. Because the UR5 does not have a spherical wrist, which would allow the use of the method of decoupling singularities, it is unclear whether this method can be applied to the UR5 manipulator. Therefore, it is important to find a method to calculate the singular configurations of the UR5. The singular configurations can then be avoided by considering them in the real-time trajectory generation and tracking, which becomes realizable when the C-API for the UR5 is acquired.



# A. Definitions and Notions

## A.1. Basic Rotation and Transformation Matrices

It is common to describe complex rotations by using basic rotations around only one axis at a time and by performing them in a consecutive order. The same approach is used for homogeneous transformations, where the basic movement is split in rotations and translations. Abbreviations for the trigonometric functions  $\cos \alpha$  and  $\sin \alpha$  are used here and throughout the thesis, i.e.,

$$\cos \alpha = c_\alpha, \quad \sin \alpha = s_\alpha \quad (\text{A.1})$$

An overview of the basic rotation and homogeneous transformation matrices is given in the following.

The set of basic rotation matrices is

$$\begin{aligned} \mathbf{R}_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix} \\ \mathbf{R}_y(\beta) &= \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \\ \mathbf{R}_z(\gamma) &= \begin{bmatrix} c_\gamma & -s_\gamma & 0 \\ s_\gamma & c_\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.2})$$

The set of basic homogeneous transformation matrices is

$$\begin{aligned}
 \text{Trans}_{x,a} &= \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{Rot}_{x,\alpha} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{Trans}_{y,b} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{Rot}_{y,\beta} &= \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{Trans}_{z,c} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{Rot}_{z,\gamma} &= \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{A.3}$$

## A.2. Skew Symmetric Matrices

Skew symmetric matrices have shown to simplify the calculation of the derivative of a rotation matrix significantly and are therefore important for the derivation of the velocity kinematics.

For a matrix  $\mathbf{S} \in \mathbb{R}^{n \times n}$  it holds that

$$\mathbf{S} := \text{skew symmetric} \iff \mathbf{S}^T + \mathbf{S} = 0 \tag{A.4}$$

If  $\mathbf{S} \in \mathbb{R}^{3 \times 3}$  is a skew symmetric matrix and a vector  $\mathbf{a} = [a_x, a_y, a_z]^T$  is given it holds that

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{A.5}$$

Furthermore it is shown in [26] that the derivative of a rotation matrix  $\mathbf{R}(\alpha)$  is equivalent to a matrix multiplication by a skew symmetric matrix  $\mathbf{S}$ , such as

$$\frac{d}{d\alpha} \mathbf{R} = \mathbf{S} \mathbf{R}(\alpha) \tag{A.6}$$

By denoting  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  as the three unit basis coordinate vectors

$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.7}$$

the derivative of the basic rotation matrices, see Equation (A.2), are

$$\frac{d}{d\alpha} \mathbf{R}_{x,\alpha} = \mathbf{S}(\mathbf{i}) \mathbf{R}_{x,\alpha} \quad (\text{A.8})$$

$$\frac{d}{d\beta} \mathbf{R}_{y,\beta} = \mathbf{S}(\mathbf{j}) \mathbf{R}_{y,\beta} \quad (\text{A.9})$$

$$\frac{d}{d\gamma} \mathbf{R}_{z,\gamma} = \mathbf{S}(\mathbf{k}) \mathbf{R}_{z,\gamma} \quad (\text{A.10})$$

Finally, for a rotation matrix  $\mathbf{R}(t)$  that describes the instantaneous orientation of a moving frame it holds that

$$\dot{\mathbf{R}}(t) = \mathbf{S}(\boldsymbol{\omega}(t)) \mathbf{R}(t) \quad (\text{A.11})$$

with

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (\text{A.12})$$

where the vector  $\boldsymbol{\omega}(t)$  is the instantaneous angular velocity of the moving frame.

### A.3. The Two-Argument Arctangent Function

The usual inverse tangent function returns an angle in the range  $(-\frac{\pi}{2}, \frac{\pi}{2})$ . In order to express the full range of angles the two-argument arctangent function is used, denoted by  $\text{Atan2}(x, y)$  [24, 26]. It is defined for all  $(x, y) \neq (0, 0)$  and equals the unique angle  $\eta$  such that

$$\cos \eta = \frac{x}{\sqrt{x^2 + y^2}}, \quad \sin \eta = \frac{y}{\sqrt{x^2 + y^2}} \quad (\text{A.13})$$

This function utilizes the signs of each argument  $x$  and  $y$  to determine which quadrant the angle  $\eta$  belongs to. This allows the correct determination of the angle  $\eta$  in a range of  $2\pi$ . If both  $x$  and  $y$  are zero, then  $\text{Atan2}$  is undefined.



## B. Software Download

The implemented software is available for download at  
<http://www.pvv.ntnu.no/~orbekk/kufieta/>

The username and password are

Username: kufieta

Password:

For more information and questions please contact  
[katharina.kufieta@gmail.com](mailto:katharina.kufieta@gmail.com)



## C. Parameter Values for the UR5 Dynamic Models

### C.1. PD Control Parameters

The PD control parameters as estimated in Section 5.1 for the model derived from the author in [21] are

$$\begin{aligned}
 \hat{\kappa}_{0,1} &= 69.0144 & \hat{\kappa}_{0,2} &= 42.2086 & \hat{\kappa}_{0,3} &= 23.9466 \\
 \hat{\kappa}_{0,4} &= 60.8311 & \hat{\kappa}_{0,5} &= 93.1483 & \hat{\kappa}_{0,6} &= 46.1087 \\
 \hat{\kappa}_{1,1} &= 69.6765 & \hat{\kappa}_{1,2} &= 42.9010 & \hat{\kappa}_{1,3} &= 24.3699 \\
 \hat{\kappa}_{1,4} &= 61.1925 & \hat{\kappa}_{1,5} &= 94.1696 & \hat{\kappa}_{1,6} &= 46.8422
 \end{aligned} \tag{C.1}$$

The PD control parameters as estimated in Section 5.3 for Model A are

$$\begin{aligned}
 \hat{\kappa}_{0,1} &= 38.1134 & \hat{\kappa}_{0,2} &= 11.2315 & \hat{\kappa}_{0,3} &= 12.9899 \\
 \hat{\kappa}_{0,4} &= 8.0265 & \hat{\kappa}_{0,5} &= 53.3125 & \hat{\kappa}_{0,6} &= 44.5480 \\
 \hat{\kappa}_{1,1} &= 38.7368 & \hat{\kappa}_{1,2} &= 11.8093 & \hat{\kappa}_{1,3} &= 31.3581 \\
 \hat{\kappa}_{1,4} &= 30.9468 & \hat{\kappa}_{1,5} &= 34.7433 & \hat{\kappa}_{1,6} &= 10.0584
 \end{aligned} \tag{C.2}$$

for Model B

$$\begin{aligned}
 \hat{\kappa}_{0,1} &= 33.9094 & \hat{\kappa}_{0,2} &= 9.5467 & \hat{\kappa}_{0,3} &= 12.9338 \\
 \hat{\kappa}_{0,4} &= 12.2111 & \hat{\kappa}_{0,5} &= 31.3581 & \hat{\kappa}_{0,6} &= 30.9468 \\
 \hat{\kappa}_{1,1} &= 34.7433 & \hat{\kappa}_{1,2} &= 10.0584 & \hat{\kappa}_{1,3} &= 13.4388 \\
 \hat{\kappa}_{1,4} &= 12.7900 & \hat{\kappa}_{1,5} &= 32.0208 & \hat{\kappa}_{1,6} &= 31.6278
 \end{aligned} \tag{C.3}$$

and for Model C

$$\begin{aligned}
 \hat{\kappa}_{0,1} &= 47.1193 & \hat{\kappa}_{0,2} &= 10.1342 & \hat{\kappa}_{0,3} &= 11.8224 \\
 \hat{\kappa}_{0,4} &= 45.1703 & \hat{\kappa}_{0,5} &= 83.5995 & \hat{\kappa}_{0,6} &= 53.4902 \\
 \hat{\kappa}_{1,1} &= 48.0059 & \hat{\kappa}_{1,2} &= 10.5656 & \hat{\kappa}_{1,3} &= 12.2736 \\
 \hat{\kappa}_{1,4} &= 46.7860 & \hat{\kappa}_{1,5} &= 84.4198 & \hat{\kappa}_{1,6} &= 54.2770
 \end{aligned} \tag{C.4}$$

## C.2. Parameters Used for the Inertia Tensor Approximation

**Table C.1.:** Cylinder parameters used for Model A and B. One cylinder is used for each link to approximate its inertia tensor. The cylinders are chosen to be homogeneous with constant mass density.

Link	# Cylinders	Mass [kg]	Height [mm]	Radius [mm]
1	1	$m_{1,cyl_1} = 3.7$	$h_{1,cyl_1} = 130$	$r_{1,cyl_1} = 58.9$
2	1	$m_{2,cyl_1} = 8.393$	$h_{2,cyl_1} = 542.8$	$r_{2,cyl_1} = 43$
3	1	$m_{3,cyl_1} = 2.275$	$h_{3,cyl_1} = 489.3$	$r_{3,cyl_1} = 37.6$
4	1	$m_{4,cyl_1} = 1.219$	$h_{4,cyl_1} = 105$	$r_{4,cyl_1} = 38.4$
5	1	$m_{5,cyl_1} = 1.219$	$h_{5,cyl_1} = 105$	$r_{5,cyl_1} = 38.4$
6	1	$m_{6,cyl_1} = 0.188$	$h_{6,cyl_1} = 35$	$r_{6,cyl_1} = 38.4$

**Table C.2.:** Inertia tensors used in Model A and B.

Link	Inertia tensor
1	$\mathbf{I}_1 = \mathbf{I}_{cyl,y}(h_1, r_1, m_1)$
2	$\mathbf{I}_2 = \mathbf{I}_{cyl,x}(h_2, r_2, m_2)$
3	$\mathbf{I}_3 = \mathbf{I}_{cyl,x}(h_3, r_3, m_3)$
4	$\mathbf{I}_4 = \mathbf{I}_{cyl,z}(h_4, r_4, m_4)$
5	$\mathbf{I}_5 = \mathbf{I}_{cyl,z}(h_5, r_5, m_5)$
6	$\mathbf{I}_6 = \mathbf{I}_{cyl,z}(h_6, r_6, m_6)$



*C.2. Parameters Used for the Inertia Tensor Approximation*

---

**Table C.3.:** Cylinder parameters used for Model C. 17 cylinders are used altogether to calculate the six inertia tensors. Each link is approximated with either one, two or five cylinders each. The cylinders are chosen to be homogeneous with constant mass density.

Link	# Cylinders	Mass [kg]	Height [mm]	Radius [mm]
1	2	$m_{1,cyl_1} = 3$	$h_{1,cyl_1} = 54.19$	$r_{1,cyl_1} = 58.9$
		$m_{1,cyl_2} = 0.7$	$h_{1,cyl_2} = 75.81$	$r_{1,cyl_2} = 58.9$
2	5	$m_{2,cyl_{1.1}} = 3$	$h_{2,cyl_{1.1}} = 54.19$	$r_{2,cyl_{1.1}} = 58.9$
		$m_{2,cyl_{1.2}} = 0.7$	$h_{2,cyl_{1.2}} = 75.81$	$r_{2,cyl_{1.2}} = 58.9$
		$m_{2,cyl_2} = 0.993$	$h_{2,cyl_2} = 307.2$	$r_{2,cyl_2} = 43$
		$m_{2,cyl_{3.1}} = 3$	$h_{2,cyl_{3.1}} = 54.19$	$r_{2,cyl_{3.1}} = 58.9$
		$m_{2,cyl_{3.2}} = 0.7$	$h_{2,cyl_{3.2}} = 75.81$	$r_{2,cyl_{3.2}} = 58.9$
3	5	$m_{3,cyl_{1.1}} = 1$	$h_{3,cyl_{1.1}} = 48.46$	$r_{3,cyl_{1.1}} = 38.4$
		$m_{3,cyl_{1.2}} = 0.219$	$h_{3,cyl_{1.2}} = 56.54$	$r_{3,cyl_{1.2}} = 38.4$
		$m_{3,cyl_2} = 0.686$	$h_{3,cyl_2} = 294.7$	$r_{3,cyl_2} = 37.6$
		$m_{3,cyl_{3.1}} = 0.1$	$h_{3,cyl_{3.1}} = 52.75$	$r_{3,cyl_{3.1}} = 58.9$
		$m_{3,cyl_{3.2}} = 0.27$	$h_{3,cyl_{3.2}} = 37.3$	$r_{3,cyl_{3.2}} = 58.9$
4	2	$m_{4,cyl_1} = 1$	$h_{4,cyl_1} = 48.46$	$r_{4,cyl_1} = 38.4$
		$m_{4,cyl_2} = 0.219$	$h_{4,cyl_2} = 56.54$	$r_{4,cyl_2} = 38.4$
5	2	$m_{5,cyl_1} = 1$	$h_{5,cyl_1} = 48.46$	$r_{5,cyl_1} = 38.4$
		$m_{5,cyl_2} = 0.219$	$h_{5,cyl_2} = 56.54$	$r_{5,cyl_2} = 38.4$
6	1	$m_{6,cyl_1} = 0.188$	$h_{6,cyl_1} = 35$	$r_{6,cyl_1} = 38.4$

C. Parameter Values for the UR5 Dynamic Models

**Table C.4.:** Inertia tensors and distances used in Model C. The distance  $\mathbf{o}_{p_{m_i,j}}^{PM_i}$  is measured from the global mass center point of link  $l_i$  to the local mass center point  $p_{m_i,j}$  of the cylinders ( $i, j$ ) in coordinate frame  $i$ .

Link	Distance $\mathbf{o}_{p_{m_i,j}}^{PM_i}$ [mm]	Local inertia tensor
1	$\mathbf{o}_{p_{m_1,1}}^{PM_1} = [ 0, -12.29, 0 ]$	$\mathbf{I}_{1,1} = \mathbf{I}_{\text{cyl},y}(h_{1,1}, r_{1,1}, m_{1,1})$
	$\mathbf{o}_{p_{m_1,2}}^{PM_1} = [ 0, 52.71, 0 ]$	$\mathbf{I}_{1,2} = \mathbf{I}_{\text{cyl},y}(h_{1,2}, r_{1,2}, m_{1,2})$
2	$\mathbf{o}_{p_{m_2,1.1}}^{PM_2} = [-212.5, 0, -16.26]$	$\mathbf{I}_{2,1.1} = \mathbf{I}_{\text{cyl},z}(h_{2,1.1}, r_{2,1.1}, m_{2,1.1})$
	$\mathbf{o}_{p_{m_2,1.2}}^{PM_2} = [-212.5, 0, 48.74]$	$\mathbf{I}_{2,1.2} = \mathbf{I}_{\text{cyl},z}(h_{2,1.2}, r_{2,1.2}, m_{2,1.2})$
	$\mathbf{o}_{p_{m_2,2}}^{PM_2} = [ 0, 0, 21.64]$	$\mathbf{I}_{2,2} = \mathbf{I}_{\text{cyl},x}(h_{2,2}, r_{2,1}, m_{2,2})$
	$\mathbf{o}_{p_{m_2,3.1}}^{PM_2} = [ 212.5, 0, -16.26]$	$\mathbf{I}_{2,3.1} = \mathbf{I}_{\text{cyl},z}(h_{2,3.1}, r_{2,3.1}, m_{2,3.1})$
	$\mathbf{o}_{p_{m_2,3.2}}^{PM_2} = [ 212.5, 0, 48.74]$	$\mathbf{I}_{2,3.2} = \mathbf{I}_{\text{cyl},z}(h_{2,3.2}, r_{2,3.2}, m_{2,3.2})$
3	$\mathbf{o}_{p_{m_3,1.1}}^{PM_3} = [-119.93, 0, 19.27]$	$\mathbf{I}_{3,1.1} = \mathbf{I}_{\text{cyl},z}(h_{3,1.1}, r_{3,1.1}, m_{3,1.1})$
	$\mathbf{o}_{p_{m_3,1.2}}^{PM_3} = [-119.93, 0, -33.23]$	$\mathbf{I}_{3,1.2} = \mathbf{I}_{\text{cyl},z}(h_{3,1.2}, r_{3,1.2}, m_{3,1.2})$
	$\mathbf{o}_{p_{m_3,2}}^{PM_3} = [ 65.82, 0, -9 ]$	$\mathbf{I}_{3,2} = \mathbf{I}_{\text{cyl},x}(h_{3,2}, r_{3,1}, m_{3,2})$
	$\mathbf{o}_{p_{m_3,3.1}}^{PM_3} = [ 272.07, 0, 17.12]$	$\mathbf{I}_{3,3.1} = \mathbf{I}_{\text{cyl},z}(h_{3,3.1}, r_{3,3.1}, m_{3,3.1})$
	$\mathbf{o}_{p_{m_3,3.2}}^{PM_3} = [ 272.07, 0, -27.9 ]$	$\mathbf{I}_{3,3.2} = \mathbf{I}_{\text{cyl},z}(h_{3,3.2}, r_{3,3.2}, m_{3,3.2})$
4	$\mathbf{o}_{p_{m_4,1}}^{PM_4} = [ 0, 0, 9.43]$	$\mathbf{I}_{4,1} = \mathbf{I}_{\text{cyl},z}(h_{4,1}, r_{4,1}, m_{4,1})$
	$\mathbf{o}_{p_{m_4,2}}^{PM_4} = [ 0, 0, -43.07]$	$\mathbf{I}_{4,2} = \mathbf{I}_{\text{cyl},z}(h_{4,2}, r_{4,2}, m_{4,2})$
5	$\mathbf{o}_{p_{m_5,1}}^{PM_5} = [ 0, 0, 9.43]$	$\mathbf{I}_{5,1} = \mathbf{I}_{\text{cyl},z}(h_{5,1}, r_{5,1}, m_{5,1})$
	$\mathbf{o}_{p_{m_5,2}}^{PM_5} = [ 0, 0, -43.07]$	$\mathbf{I}_{5,2} = \mathbf{I}_{\text{cyl},z}(h_{5,2}, r_{5,2}, m_{5,2})$
6	$\mathbf{o}_{p_{m_6,1}}^{PM_6} = [ 0, 0, 0 ]$	$\mathbf{I}_6 = \mathbf{I}_{\text{cyl},z}(h_6, r_6, m_6)$

## D. Acronyms and List of Symbols

### Acronyms

AC	Alternating current
A/D	Analog/Digital
API	Application programming interface
C, C++	Programming languages
CAN	Controller area network
CMT	Communication MT, communication protocol of the Xsens MTi accelerometer
CSV	Comma-separated values
D	Dimension
DC	Direct current
DH	Denavit-Hartenberg, a method used to model manipulators
DOF	Degrees of freedom
EKF	Extended Kalman filter
F/T	Force/Torque
IMU	Inertial measurement unit
IP	Internet protocol
MAE	Mean absolute error
MAT	A file format from Matlab
MEMS	Micro-electro-mechanical system
NLS	Nonlinear least squares
NLSE	Nonlinear least squares estimator
PC	Personal computer
PD	Proportional-Derivative controller
PID	Proportional-Integral-Derivative controller
PoE	Power over ethernet
PTP	Point to point
SDK	Software development kit
TCP	Transmission control protocol

UDP	User datagram protocol
UR5	A robot from Universal Robots, Model UR5
USB	Universal serial bus

### Calligraphic Letters

$\mathcal{I}$	Inertia tensor, expressed in the inertial frame $o_0x_0y_0z_0$ , $\mathcal{I} \in \mathbb{R}^{3 \times 3}$
$\mathcal{J}(\cdot)$	Cost function for optimization problems
$\mathcal{K}$	Kinetic energy of a rigid body

### Greek Capital Letters

$\Pi_B$	Plane acting as a border between two cylinders, that together form a bigger cylinder.
$\Omega_c$	Cutoff frequency

### Greek Lowercase Letters

$\alpha_i$	Link twist of link $i$ , DH parameter
$\beta$	Minimal representation for the orientation of the end effector frame relative to the inertial frame, $\beta \in \mathbb{R}^3$
$\delta$	Dead zone threshold
$\epsilon$	Position modeling error, $\epsilon \in \mathbb{R}^n$
$\epsilon_F$	Force modeling error from force estimation based on torque model without friction modeling, $\epsilon_F \in \mathbb{R}^n$
$\epsilon_{F^*}$	Force modeling error from force estimation based on torque model with friction modeling, $\epsilon_{F^*} \in \mathbb{R}^n$
$\epsilon_{F^u}$	Force modeling error from force estimates coming from the UR5, $\epsilon_{F^u} \in \mathbb{R}^n$
$\epsilon_{\hat{\tau}}$	Torque modeling error, $\epsilon_{\hat{\tau}} \in \mathbb{R}^n$
$\epsilon_{\ddot{x}}$	End effector acceleration modeling error, $\epsilon_{\ddot{x}} \in \mathbb{R}^n$

---

$\theta$	Vector with joint position values for revolute joints, $\theta \in \mathbb{R}^n$
$\theta_i$	Joint angle of link $i$ , DH parameter
$\vartheta$	Second Euler angle, see $\phi$
$\kappa$	Vector containing the diagonal control parameters of the PD gain matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ , $\kappa = [\kappa_0, \kappa_1] \in \mathbb{R}^{2n}$
$\lambda$	Angle of the axis/angle representation of a rotation, see $\mathbf{R}_{l,\lambda}$ , $\lambda \in \mathbb{R}$
$\xi_j^i$	Linear and angular velocity vector of link $j$ , expressed in coordinate frame $i$ , based on a non-minimal representation for the orientation of link $j$ , $\xi_j^i \in \mathbb{R}^6$
$\rho$	Mass density of a body
$\varrho$	Friction coefficient
$\tau$	Joint torque vector, $\tau \in \mathbb{R}^n$
$\tau_{cou}$	Coulomb friction torque vector, $\tau_{cou} \in \mathbb{R}^n$
$\tau_{ext}$	Vector of joint torques that are induced by an external force, $\tau_{ext} \in \mathbb{R}^n$
$\tau_{fr}$	Friction torque vector, $\tau_{fr} \in \mathbb{R}^n$
$\tau_{vis}$	Viscous friction torque vector, $\tau_{vis} \in \mathbb{R}^n$
$\tau_\mu$	Vector of joint torques corresponding to the manipulator dynamics, $\tau_\mu \in \mathbb{R}^n$
$\nu_c$	Coulomb friction coefficient vector, $\nu_c \in \mathbb{R}^n$
$\nu_v$	Viscous friction coefficient vector, $\nu_v \in \mathbb{R}^n$
$\phi$	Orientation vector based on Euler angles, $\phi = [\varphi, \vartheta, \psi]^T$
$\varphi$	First Euler angle, see $\phi$
$\psi$	Third Euler angle, see $\phi$
$\omega_j^i$	Angular velocity vector of link $j$ , expressed in coordinate frame $i$ , based on a non-minimal representation of the orientation of link $j$ , $\omega_j^i \in \mathbb{R}^3$
$\omega_n$	Natural frequency

### Latin Capital Letters

$\mathbf{A}_i$	Homogeneous transformation matrix that gives position and orientation of coordinate frame $i$ with respect to frame $i - 1$ , $\mathbf{A}_i \in \mathbb{R}^{4 \times 4}$
$\text{Atan2}(x, y)$	Arctangent of the ratio $\frac{x}{y}$
$\mathbf{B}$	Transformation matrix between the non-minimalistic angular velocity $\boldsymbol{\omega}$ and the minimalistic angular velocity $\dot{\boldsymbol{\beta}}$ , $\mathbf{B} \in \mathbb{R}^{3 \times 3}$
$C$	Centroid of a body
$\mathbf{C}$	Centrifugal & coriolic matrix, $\mathbf{C} \in \mathbb{R}^{n \times n}$
$\mathbf{D}$	Inertia matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$
$\mathbf{F}$	End effector forces, $\mathbf{F} = [F_x, F_y, F_z, n_x, n_y, n_z] \in \mathbb{R}^6$
$F_C$	Coulomb force
$F_{fc}$	Coulomb friction
$F_{fv}$	Viscous friction
$F_N$	Load
$F_\chi$	Forces acting on the end effector in direction of the $\chi$ -axis, for $\chi \in \{x, y, z\}$ , see $\mathbf{F}$
$\hat{\mathbf{F}}$	Estimated end effector forces based on torque model without friction modeling, $\hat{\mathbf{F}} \in \mathbb{R}^6$
$\hat{\mathbf{F}}^*$	Estimated end effector forces based on torque model with friction modeling, $\hat{\mathbf{F}}^* \in \mathbb{R}^6$
$\hat{\mathbf{F}}^u$	Estimated end effector forces received from the UR5, $\hat{\mathbf{F}}^u \in \mathbb{R}^6$
$\mathbf{H}$	Homogeneous transformation matrix, $\mathbf{H} \in \mathbb{R}^{4 \times 4}$
$H(\Omega)$	Frequency response of butterworth filter
$\mathbf{I}_C$	Inertia tensor, expressed in the centroid attached frame $o_C x_C y_C z_C$ , $\mathbf{I}_C \in \mathbb{R}^{3 \times 3}$
$\mathbf{I}_{\text{cyl}, \cdot}$	Inertia tensor of a cylinder with constant mass density, where $\cdot$ denotes the rotational axis of the cylinder, $\mathbf{I}_{\text{cyl}, \cdot} \in \mathbb{R}^{3 \times 3}$
$\mathbf{I}_M$	Inertia tensor, expressed in the body attached frame $o_M x_M y_M z_M$ , $\mathbf{I}_M \in \mathbb{R}^{3 \times 3}$

---

$\mathbf{J}$	Manipulator Jacobian, $\mathbf{J} \in \mathbb{R}^{6 \times n}$
$\mathbf{J}_a$	Analytical Jacobian, based on minimalistic representation of the end effector position, $\mathbf{J}_a \in \mathbb{R}^{6 \times n}$
$\mathbf{J}_g$	Geometric Jacobian, based on non-minimalistic representation of the end effector position, $\mathbf{J}_g \in \mathbb{R}^{6 \times n}$
$\mathbf{J}_{M_j}$	Jacobian for the mass center point $p_M$ on link $l_j$ , $\mathbf{J}_{M_j} \in \mathbb{R}^{6 \times n}$
$\mathbf{J}_{P_j}$	Jacobian for an arbitrary point $P$ on link $l_j$ , $\mathbf{J}_{P_j} \in \mathbb{R}^{6 \times n}$
$\mathbf{J}_v$	Linear part of geometric Jacobian, $\mathbf{J}_v \in \mathbb{R}^{3 \times n}$
$\mathbf{J}_\omega$	Angular part of geometric Jacobian, $\mathbf{J}_\omega \in \mathbb{R}^{3 \times n}$
$K$	Kinetic energy for an n-link manipulator
$\mathbf{K}_0$	Proportional gain matrix, $\mathbf{K}_0 \in \mathbb{R}^{n \times n}$
$\mathbf{K}_1$	Derivative gain matrix, $\mathbf{K}_1 \in \mathbb{R}^{n \times n}$
$M_i$	Mass of link $i$
$N$	Number of measurement samples
$N_b$	Filter order
$P$	Potential energy for an n-link manipulator
$\mathbf{R}$	Rotation matrix, $\mathbf{R} \in SO(n)$ , $\mathbf{R}^T = \mathbf{R}^{-1}$ , $\det \mathbf{R} = 1$
$\mathbf{R}_{l,\lambda}$	Axis/angle transformation matrix, $\mathbf{R}_{l,\lambda} \in SO(n)$
Rot.,.	Basic homogeneous transformation matrix describing a pure rotation about a given axis with a given angle, $\text{Rot.,.} \in \mathbb{R}^{4 \times 4}$
$\mathbf{R}_{ZYZ}$	ZYZ-Euler angle transformation matrix, $\mathbf{R}_{ZYZ} \in SO(n)$
$\mathbf{S}$	Skew symmetric matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$
$SO(n)$	Special Orthogonal group of order $n$
$\mathbf{T}_j^i$	Homogeneous transformation matrix that gives position and orientation of frame $j$ with respect to frame $i$ , $\mathbf{T}_j^i \in \mathbb{R}^{4 \times 4}$
Trans.,.	Basic homogeneous transformation matrix describing a pure translation in direction of a given axis with a given distance, $\text{Trans.,.} \in \mathbb{R}^{4 \times 4}$
$\mathbf{X}$	End effector pose based on a minimal representation for the orientation of the end effector frame, relative to the inertial frame, $\mathbf{X} \in \mathbb{R}^6$

### Latin Lowercase Letters

$a_i$	Link length of link $i$ , DH parameter
$\mathbf{a}_q$	Control input for double integrator system in joint space, $\mathbf{a}_q \in \mathbb{R}^n$
$\mathbf{a}_X$	Control input for double integrator system in task space, $\mathbf{a}_X \in \mathbb{R}^n$
$\mathbf{b}$	Offset of the linear function describing the current-torque relationship, $\mathbf{b} \in \mathbb{R}^n$
$c_{ijk}$	Christoffel symbols
$c_\theta$	Cosine of $\theta$
$\mathbf{d}$	Translation vector, $\mathbf{d} \in \mathbb{R}^3$
$d_i$	Link offset of link $i$ , DH parameter
$d_{ij}$	$(i, j)$ -th element of the inertia matrix $\mathbf{D}$
$\mathbf{g}$	Gravity Vector, $\mathbf{g} \in \mathbb{R}^n$
$g_k$	$(k)$ -th element of the gravity vector $\mathbf{g}$
$\mathbf{g}_0$	Vector giving the direction of gravity in the inertial frame $o_0x_0y_0z_0$ , $\mathbf{g}_0 \in \mathbb{R}^3$
$h$	Height of a cylinder
$\mathbf{i}$	Currents vector, $\mathbf{i} \in \mathbb{R}^n$
$\mathbf{i}_a$	Armature currents vector, $\mathbf{i}_a \in \mathbb{R}^n$
$j_i$	Joint $i$
$\mathbf{k}$	Unit base coordinate vector in z-direction, $\mathbf{k} \in \mathbb{R}^3$
$\mathbf{k}_\tau$	Torque constants vector, $\mathbf{k}_\tau \in \mathbb{R}^n$
$\mathbf{l}$	Axis of the axis/angle representation of a rotation, see $\mathbf{R}_{\mathbf{l}, \lambda}$ , $\mathbf{l} \in \mathbb{R}^3$
$l_i$	Link $i$
$m$	Mass of a body
$n$	Number of joints
$n_\chi$	Torques acting on the end effector around the $\chi$ -axis, for $\chi \in \{x, y, z\}$ , see $\mathbf{F}$
$\mathbf{o}_j^i$	Translation vector pointing from the origin of coordinate frame $i$ to the origin of coordinate frame $j$ , $\mathbf{o}_j^i \in \mathbb{R}^3$
$\mathbf{o}_{pC}^{pM}$	Translation vector pointing from the mass center point to the centroid of a body, $\mathbf{o}_{pC}^{pM} \in \mathbb{R}^3$
$o_0x_0y_0z_0$	Inertial/Base coordinate frame, a cartesian coordinate frame attached to the robot base.
$o_Cx_Cy_Cz_C$	Cartesian coordinate frame attached to the centroid of a body



---

$o_i x_i y_i z_i$	Cartesian coordinate frame $i$ , attached to link $i$
$o_M x_M y_M z_M$	Body attached frame, a cartesian coordinate frame attached to the mass center point of a body
$p$	Point $p$ in a room
$\mathbf{p}_C$	Vector pointing to the centroid of a body, $\mathbf{p}_C \in \mathbb{R}^3$
$\mathbf{p}^i$	Vector pointing to a point $p$ , expressed in frame $i$ , $\mathbf{p}^i \in \mathbb{R}^3$
$p_M$	Mass center point
$\mathbf{p}_M$	Vector pointing to the mass center point of a body, $\mathbf{p}_M \in \mathbb{R}^3$
$\mathbf{q}$	Joint position variables vector, $\mathbf{q} \in \mathbb{R}^n$
$r$	Radius of a cylinder
$r_{ij}$	$(i, j)$ -th element of a rotation matrix $\mathbf{R}$
$\mathbf{r}_{Mi}^i$	Vector pointing to the mass center point of a link $i$ , expressed in the coordinate frame $i$ , $\mathbf{r}_{Mi}^i \in \mathbb{R}^3$
$\mathbf{r}_{Pj}^i$	Vector pointing to a point $p$ on a link $j$ , expressed in the coordinate frame $i$ , $\mathbf{r}_{Pj}^i \in \mathbb{R}^3$
$s_\theta$	Sine of $\theta$
$t$	Continuous time
$t_k$	Time instances
$\mathbf{u}$	Control input, $\mathbf{u} \in \mathbb{R}^n$
$\mathbf{v}_j^i$	Linear velocity vector of link $j$ in coordinate frame $i$ , $\mathbf{v}_j^i \in \mathbb{R}^3$
$x_B$	Point where the border plane $\Pi_B$ between two cylinders crosses the x-axis
$\mathbf{y}$	Available measurements, $\mathbf{y} \in \mathbb{R}^{2n}$
$\mathbf{z}_i$	Axis of rotation of a revolute joint $j_i$ , $\mathbf{z}_i \in \mathbb{R}^3$

## Special Characters

$\forall$	For all
$\mathbf{i}$	Unit base coordinate vector in x-direction, $\mathbf{i} \in \mathbb{R}^3$
$\mathbf{j}$	Unit base coordinate vector in y-direction, $\mathbf{j} \in \mathbb{R}^3$
$(\cdot)_j^i$	Notation, from coordinate frame $i$ to coordinate frame $j$ , expressed in coordinate frame $i$
$\hat{(\cdot)}$	Notation, estimated/modeled value
$(\cdot)^d$	Notation, desired or reference value
$\oplus$	Logic symbol XOR
$\wedge$	Logic symbol AND



## Eidesstattliche Erklärung

Ich versichere hiermit, dass ich, Katharina Kufieta, die vorliegende Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäß entlehnte Stellen als solche kenntlich gemacht habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

---

Ort, Datum

---

Unterschrift



## Bibliography

- [1] L.M. Aksman, C.R. Carignan, and D.L. Akin. Force estimation based compliance control of harmonically driven manipulators. *IEEE International Conf. on Robotics and Automation*, pages 4208–4213, April 2007.
- [2] ATI Industrial Automation. *Network Force/Torque Sensor System, Compilation of Manuals*, February 2013.
- [3] P. Axelsson, R. Karlsson, and M. Norrlöf. Bayesian state estimation of a flexible industrial robot. *Control Engineering Practice*, 20(11):1220–1228, November 2012.
- [4] P. Axelsson and M. Norrlöf. Method to estimate the position and orientation of a triaxial accelerometer mounted to an industrial manipulator. *10th International IFAC Symposium on Robot Control*, 2012.
- [5] P.I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.
- [6] O. Egeland and J.T. Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics AS, 2002.
- [7] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Sensor fusion of force and acceleration for robot force control. *Proc. of 2004 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 3009–3014, 2004.
- [8] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Automatic calibration procedure for a robotic manipulator force observer. *Proc. of 2005 IEEE International Conf. on Robotics and Automation*, pages 2703–2708, April 2005.
- [9] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Force and acceleration sensor fusion for compliant robot motion control. *Proc. of 2005 IEEE International Conf. on Robotics and Automation*, pages 2709–2714, April 2005.

- [10] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Self calibrating procedure for a 3D force observer. *Proc. of 44th IEEE Conf. on Decision and Control, and the European Control Conf. 2005*, pages 6812–6817, December 2005.
- [11] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Generalized contact force estimator for a robot manipulator. *Proc. of 2006 IEEE International Conf. on Robotics and Automation*, pages 4019–4024, May 2006.
- [12] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Sensor fusion for compliant robot motion control. *IEEE Transactions on Robotics*, 24(2):430–441, 2008.
- [13] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson. Self-calibrated robotic manipulator force observer. *Robotics and Computer-Integrated Manufacturing*, 25(2):366–378, April 2009.
- [14] Daniel Helmick, Avi Okon, and Matt Dicicco. A comparison of force sensing techniques for planetary manipulation. Technical report, Jet Propulsion Laboratory, 4800 Oak Grove Dr, Pasadena CA 91109, October 2005.
- [15] H. Høifødt. Dynamic modeling and simulation of robot manipulators: The newton-euler formulation. Master’s thesis, Norwegian University of Science and Technology, NTNU, 2011.
- [16] J. Jung, J. Lee, and K. Huh. Robust contact force estimation for robot manipulators in three-dimensional space. *Proc. of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, pages 1317–1327, September 2006.
- [17] T. Kröger, D. Kubus, and F.M. Wahl. 6D force and acceleration sensor fusion for compliant manipulation control. *Proc. of 2006 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 2626–2631, October 2006.
- [18] H.P. Langtangen. *Python Scripting for Computational Science*. Springer, 2004.
- [19] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms & Applications*. Pearson Prentice Hall, fourth edition, 2007.

- [20] M. Quigley, R. Brewer, S.P. Soundararaj, V. Pradeep, and A.Y. Ng. Low-cost accelerometers for robotic manipulator perception. *2010 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 6168–6174, October 2010.
- [21] M.R.P. Ragazzon. Robot manipulator collision handling in unknown environment without using external sensors. Master’s thesis, Norwegian University of Science and Technology, NTNU, 2012.
- [22] P. Roan, N. Deshpande, Y. Wang, and B. Pitzer. Manipulator state estimation with low cost accelerometers and gyroscopes. *2012 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 4822–4827, October 2012.
- [23] J. Schrimpf. *Sensor-based Real-time Control of Industrial Robots*. PhD thesis, Norwegian University of Science and Technology, NTNU, 2013.
- [24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning & Control*. Springer, 2009.
- [25] J.W.L. Simpson, C.D. Cook, and Z. Li. Sensorless force estimation for robots with friction. *Proc. of 2002 Australasian Conf. on Robotics and Automation*, pages 94–99, November 2002.
- [26] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.
- [27] M.K. Steven. *Fundamentals of Statistical Signal Processing, Vol. 1: Estimation Theory*. Prentice Hall PTR, 1993.
- [28] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Force controlled robotic assembly without a force sensor. *2012 IEEE International Conf. on Robotics and Automation*, pages 1538–1543, May 2012.
- [29] S. Tachi, T.M. Sakaki, H. Arai, S. Nishizawa, and J.F. Pelaez-Polo. Impedance control of a direct-drive manipulator without using force sensors. *Advanced Robotics*, pages 183–205, July 1991.
- [30] Universal Robots. *PolyScope Manual*, February 2013.
- [31] Universal Robots. *The URScript Programming Language*, February 2013.

- [32] Universal Robots. *User Manual: UR5*, February 2013.
- [33] M. Van Damme, P. Beyl, B. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, a. Matthys, and D. Lefeber. Estimating robot end-effector force from noisy actuator torque measurements. *2011 IEEE International Conf. on Robotics and Automation*, pages 1108–1113, May 2011.
- [34] Y. Wang, W. Chen, and M. Tomizuka. Extended kalman filtering for robot joint angle estimation using MEMS inertial sensors. *Proc. of 6th IFAC Symposium on Mechatronic Systems*, pages 406–413, April 2013.
- [35] Xsens Technologies B.V. *MT Low-Level Communication Protocol Documentation*, October 2010.
- [36] Xsens Technologies B.V. *MT Manager User Manual*, October 2010.
- [37] Xsens Technologies B.V. *MT Software Development Kit Documentation*, October 2010.
- [38] Xsens Technologies B.V. *MTi and MTx User Manual and Technical Documentation*, October 2010.