AUSTRIAN
MARSHALL PLAN FOUNDATION

Marshall Plan Scholarship Research Report[1]

Robust Control for Modular Robotics

Author

Martin Winkler

20th October, 2016

[1] The content of this report is based on the authors master thesis with same title.

Abstract

To effectively use force control for real world tasks like robotic manipulation or locomotion, the determination of an accurate model of the robot's kinematics and dynamics is needed. If modular robots are to match the performance of purpose-built systems, this process needs to be automated or heavily assisted. In this thesis, the primary task is to implement algorithms that allow quick reconfiguration and parameter estimation of modular robots using their distributed position, force, and inertial sensors. It is identified if the robot topology, kinematic configuration and dynamic parameters can be figured out with the given hardware and proper algorithms. All of this work was evaluated with Carnegie Mellon Universities Biorobotic labs X5-series modules, so practical realities of noisy sensors, imperfect torque measurements, and other hardware effects like network latency, jitter and other disturbances are taken into account. Results show, that kinematic parameter identification yields good results on the robot manipulators, but vertical motions against gravity are very prone to noise. Dynamic parameter identification is evaluated by calculating supplementary torques from the estimated parameters to supply them as additional input command for the actuators. Trajectory following was improved in all cases.

Table of Contents

Table of Figures

List of Tables

1. Introduction

It has been over a century since the Czech writer Karel Čapek coined the term *robot* in his play R.U.R. in 1920 [1]. Since then, robots are all around us, and especially in recent decades they have intruded not just corporate, taking our jobs, but also personal life. Cleaning robots, automatic lawn mowers or self-driving cars from Google and others have found their ways into everyday life. Furthermore, a lot of work and ambition is shown through DARPA's driverless cars and robotics challenges[2], or ambitious goals like space exploration, especially upcoming Mars projects from SpaceX[3] and NASA[4]. Being faced with our aging populace, Toyota is investing not just into self-driving cars, but also into robot helpers for households [5].

In general, robots used to be developed for special applications over their operational lifetime. They are developed for well-defined tasks, and perform well under those constraints. With the advance of robotics, the trend is also to use robots in unpredictable and changing environments like on the International Space Station (ISS). While robot manipulators can be reprogrammed for new tasks, the given hardware configuration limits the number of applications.

Modular Robotics is an approach where independent hardware components can be quickly assembled for individual tasks, just like combining Lego bricks. This yields benefits during development of robot systems and can also lead to agile manufacturing, where a different range of tasks can be solved by quickly adapting and reconfiguring the robot to solve the task at hand.

To effectively perform robot manipulation for real world tasks like force control or locomotion, an accurate model of the robot's kinematics and dynamics is needed. The parameters for the models are normally calculated by hand and are thus time consuming and error prone manual processes, which are done for each robot configuration specifically. To allow for quick reconfiguration, this process needs to be automated as much as possible to match the performance of purpose-built systems.

Another problem addressed, is that through wear or deformation, the joints and links change its parameters like length, leading to inaccurate control. Automatic parameter and model identification allows for a fast reassessment of parameters without the need for a time consuming and complex disassemble and reevaluation of the robot.

Allowing the robot to identify its own kinematic and dynamic model from scratch, or to update its parameters repeatedly when needed, can lead to new applications and advantages over the life time of a robot. This can allow for:

- Self-reconfiguration
- Self-repairing
- Self-assembling

[2] http://www.darpa.mil/
[3] http://www.spacex.com/
[4] https://www.nasa.gov/
[5] http://www.toyota.com/

- Error detection
- Error handling

1.1 Research purpose

Given the unique torque controlled modular research platform developed in the Biorobotics lab at Carnegie Mellon University, can kinematic and dynamic identification schemes be used to reliably identify required parameters for the control of the system? Specifically, how well does identification fair with the modules, given that the Inertial Measurement Units (IMUs) are not calibrated. Which robot configurations and positions work and which identification trajectories are needed for the best results. Parameters of interest include kinematics parameters like rotation axis and length of links and dynamic parameters like inertia, or weight to calculate motor torques.

1.2 Outline

Chapter 2 introduces important topics which form the basis of the following work. Robot manipulators, which are the key part of this thesis, are explained. A more uncommon rotation approach, the axis-angle representation, is elaborated. Important kinematic and dynamic details, which are necessary for the understanding of this thesis are outlined. Finally, the hardware platform which is used for experiments is introduced.

Research done in the areas of parameter identification are outlined in chapter 3. This forms a basis for discussion about techniques in the thesis. Chapter 4 and 5 introduces the methods which are then used in the experiments in chapter 6.

Finally, chapter 7 will list conclusions to kinematic and dynamic parameter identification on the used hardware platform, and future open work.

2. Background

Robot manipulators and their properties are an important part of this thesis and serve as hardware platform for experiments. To use such a manipulator, one must know the physics behind them, which are explained by kinematics and dynamics. The following sections will give an overview of the necessary background information required for the work of this thesis.

2.1 Robot Manipulators

Robots come in many forms like humanoid walking robots, robot arms, or mobile robots like self-driving cars. Robots relevant for this thesis are mechanical structures which consist of rigid bodies, called links, connected by means of articulation, so called joints. The combination of links and joints form a serial or open kinematic chain, which often resembles an arm. In a final device, an additional end-effector is mounted to interact with the world and manipulate the environment. The mobility of the arm is guaranteed by the joints which can be either prismatic, or revolute as shown in Figure 1 [2]. Prismatic joints allow for relative translational motion between two links; revolute joints produce relative rotational motion between to links. Each joint adds a degree of freedom (DOF) to the system.

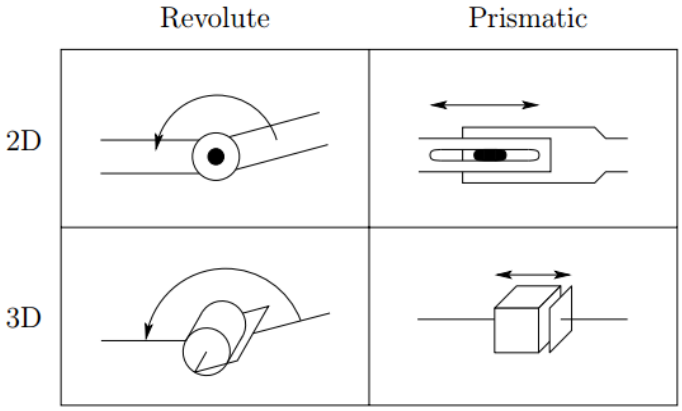

Figure 1 - Revolute and Prismatic Joint Symbols [2]

To position and orient objects in 3D space, 6-DOF robot is needed. Figure 2 displays a 6-DOF industrial robot, the KUKA KR-16 [3]. They consist of the arm for positioning, and a wrist for orienting the end-effector. These types of robots are anthropomorphic robot structures and are the most common robot manipulator type worldwide. Other types are Cartesian, cylindrical, spherical, or SCARA. For example, in Figure 3, a Cartesian manipulator is realized by a gantry structure. These manipulators allow to lift heavy weight and are used for material handling and assembly. Information about the other structures can be found in Robotics - Modelling, Planning and Control by Siciliano et al. [4].

Figure 2 - KUKA KR-16 6-DOF Industrial Robot [3]

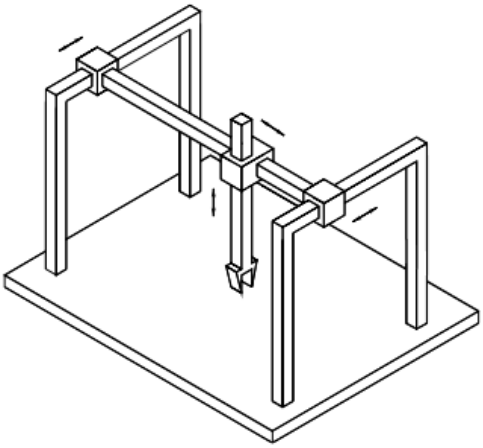

Figure 3 - Cartesian Manipulator [4]

All presented manipulators have an open kinematic chain. Closed kinematic chains are used when higher stiffness is required to allow for heavier payloads and still have good positioning accuracy. An anthropomorphic manipulator can be made into a closed kinematic chain by equipping a parallelogram geometry between shoulder and elbow joints (Figure 4) [4].

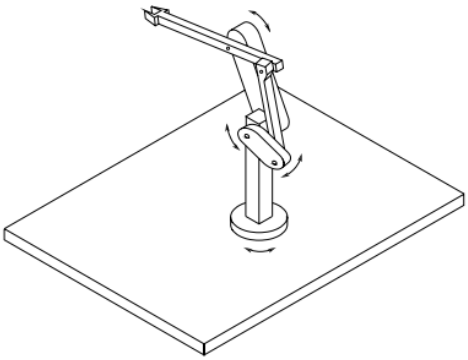

Figure 4 - Anthropomorphic Manipulator with Parallelogram [4]

2.2 Axis-Angle Representation

The axis-angle representation of rotations is used to rotate vectors around a given axis. It is based on the Rodrigues Rotation formula [5], which is also called the axis-angle rotation formula. Normally, in linear algebra a vector v is rotated around the origin of a coordinate system with a rotation matrix *R* to calculate its new position *v'* as in Equation 1:

$$v' = Rv \tag{1}$$

A different way for rotations is using the axis-angle representation which only works in 3D space. This is used for rotating about an axis which is not centered at the origin and when rotations are not performed around a standard *x*-, *y*-, and *z*-axis. The axis of rotation is a unit vector *k*, which a vector *v* rotates around by an angle θ. Figure 5 shows vector *v*, which is rotated by 180 degrees around axis *k*, resulting in vector *v'*. The vectors $v_{\parallel}$ and $v_{\perp}$ are the projections of *v* and are used for the Rodrigues formula [6].

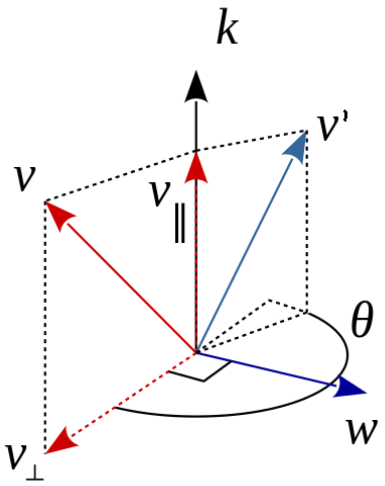

Figure 5 - Axis-Angle Rotation

2.3 Kinematics

Kinematics describes objects in motion (position, velocity & acceleration) without considering causes (why is the object moving) and the forces involved. A robot system (single arm manipulator, humanoid robot, etc.) can be described as a kinematic chain of links, connected by revolute or prismatic joints. These links are rigid bodies, and their movements affect the whole chain. Each link is dependent on the previous link, and through vector and matrix algebra these systems can be described with respect to a reference frame. This allows to determine end-effector positions and orientations to interact with the environment. This mathematical description of a robot system allows the end-effector position to be described by its joint angles. The resulting kinematic descriptions for a robot system can then be used to calculate the equations for dynamics and control [4] [7].

Kinematic parameters are for example the length of links connecting two joints, or the axis of rotation for the joints. There exist two important mathematical procedures to calculate kinematic properties which are forward kinematics and inverse kinematics.

2.3.1 Forward Kinematics

Forward kinematics, also called direct kinematics, calculates the end-effector position, given the joint properties (angles of revolute joints or displacements of prismatic joints). It maps the joint space to the operational space.

A robot manipulator forms a kinematic chain where one end is constrained to a base. At the other end an end-effector is mounted to interact with the world in the operational space. To calculate a mapping starting from the base, going from joint to joint up to the end-effector, several approaches are possible. For simple manipulators, a geometric approach can be used to analyze the structure. For more complex structures a structured recursive approach called Denavit-Hartenberg convention exists.

2.3.1.1 Denavit-Hartenberg

A common method to describe the kinematic properties of a manipulator is to use the Denavit-Hartenberg (DH) notation [8]. The modified parameters differ in the locations of coordinate's system attachment to the links and the order of the performed transformations. A basic setup of the convention with its coordinate frames and parameters is shown in Figure 6.

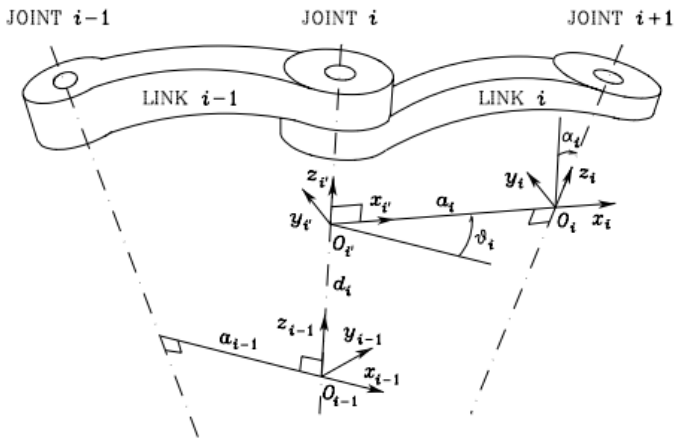

Figure 6 - DH-Parameters

Starting from the base, each joint of the manipulator is assigned a reference frame such that each joint and link can be described with regards to the previous one in the kinematic chain. The end-effector can be positioned and oriented using a different combination of joint values.

Four parameters are used for each joint and link *i*:

- Θ_i: joint angle, required rotation of x_{i-1} about the z_{i-1} to become parallel to the x_i axis
- a_i: link length, distance between z_{i-1} and z_i along the common normal between the two axes (axis x_{i-1})
- d_i: link offset, distance between x_{i-1} and x_i along the z_{i-1} axis
- α_i: link twist, required rotation of the z_{i-1} axis about the x_i axis to become parallel to the z_i axis

These parameters are then used in homogenous transformations to calculate the transformation from frame to frame. One homogenous transformation is represented by four basic transformations:

$$A_i = Rot_{z,\Theta_i} Trans_{z,d_i} Trans_{a_i} x, Rot_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\Theta_i} & -s_{\Theta_i} & 0 & 0 \\ s_{\Theta_i} & c_{\Theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$= \begin{bmatrix} c_{\Theta_i} & -s_{\Theta_i} c_{\alpha_i} & s_{\Theta_i} s_{\alpha_i} & a_i c_{\Theta_i} \\ s_{\Theta_i} & c_{\Theta_i} c_{\alpha_i} & -c_{\Theta_i} s_{\alpha_i} & a_i s_{\Theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After every A_i is calculated, the end-effector position and orientation can be calculated given the joint values. A detailed procedure can be found in [4].

2.3.2 Inverse Kinematics

In contrast to forward kinematics, inverse kinematics calculates joint values given the desired end-effector position and orientation. A mapping from the operational space to the joint space. This is important to execute desired end-effector motions in space.

Calculating the inverse kinematics is more complicated than the forward kinematics for several reasons:

- Equations are often non-linear and a closed-form solution might not exist
- Multiple solutions may exist
- In case of redundant manipulators, infinite solutions can exist
- There might be no permissible solution, depending on the manipulator kinematic structure

If finding a closed-form solution is too hard, numerical solution techniques can be used. A detailed description of how to solve inverse kinematic problems is out of the scope for this thesis. A more detailed explanation can be found in [7] and [4].

2.4 Dynamics

The dynamical model describes the relationship between the motion of the manipulator and the internal and external forces and torques that arise during its operation, e.g. inertial, centrifugal, coriolis, gravitational and actuating torques or forces. By computing forces and torques required for desired motions on a manipulator, one can verify that the physical hardware, like the joints, can produce the torques and forces needed.

The resulting dynamic equations for a robot system can then be used to simulate motion, analysis of manipulator structures and the calculation of the equations for control. Especially

simulations of the robot allow for testing prior to the use on a real robot. Two common formulations exist for the derivation of the dynamic equations [4] [7]:

- The Newton-Euler formulation is a recursive and computationally faster approach, which provides dynamic equations for calculating required actuator forces and torques
- The Lagrange formulation yields the required differential equations which determine the actuator forces and torques.

Properties for dynamics include mass, center of gravity or the inertia tensor described in chapter 2.4.1.

2.4.1 Moment of Inertia

Given an object with mass m and an acceleration a, a force F is needed to accelerate the object in a straight line which is given by Newton's law of motion:

$$F = ma \tag{3}$$

To achieve the same for rotations, a torque τ is computed, given the moment of inertia *I* and the angular acceleration α:

$$\tau = I\alpha \tag{4}$$

Every rigid body has a moment of inertia, also called rotational inertia, which is dependent on the mass distribution and axis chosen. This rotational inertia is used to calculate torques needed to create angular accelerations around a rotational axis. Larger moments result in higher torques to change the body's rotation.

Bodies rotating in three dimension can be characterized by a symmetric, positive-definite 3x3 matrix called inertia tensor, or inertia matrix. The matrix consists of the moments of inertia and products of inertia about the coordinate axes:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{5}$$

I_{xx}, I_{yy}, I_{zz} in Equation 5 are the moments of inertia about the x, y, and z axis. The integral is taken over the whole volume of the object, summing the squared distance of each particle along the axes, multiplying by the mass of the object. This results in higher inertia if most of the object is farther away from the axis of interest. This phenomenon can be observed in figure skating. When spinning on ice, rotation is faster when bringing arms closer to the body.

$$I_{xx} = m \iiint_V (y^2 + z^2) dv \tag{6}$$

$$I_{yy} = m \iiint_V (x^2 + z^2) dv \tag{7}$$

$$I_{zz} = m \iiint_V (x^2 + y^2) dv \tag{8}$$

I_{xy}, I_{xz}, I_{yx}, I_{yz}, I_{zx}, and I_{zy} in Equation 5 are the products of inertia and are a measure of symmetry. If an object is symmetric, every point on each side cancels out, and then off-diagonal elements of the inertia tensor are zero.

$$I_{xy} = I_{yx} = -m \iiint_V (xy)dv \tag{9}$$

$$I_{yz} = I_{zy} = -m \iiint_V (yz)dv \tag{10}$$

$$I_{zx} = I_{xz} = -m \iiint_V (zx)dv \tag{11}$$

If an object is symmetric, every point on one side of each axis is going to "cancel out" the corresponding point on the other side of the axis. This describes the phenomenon described above [7] [9].

2.5 Snake Robots

A major focus in the CMU Biorobotics lab at Carnegie Mellon University is the focus on highly articulated systems like snake robots [10]. The laboratory, led by Prof. Howie Choset, has underwent several iterations of various kinds of robotic snakes. Exploiting its many internal degrees of freedom, such a robot can access areas which are unreachable for humans or other kinds of wheeled, legged or flying robots like hexapods or quadcopters. Challenges addressed in the lab are design and multidimensional path planning using novel gaits. This allows the robots to use its many degrees of freedom to operate in a variety of areas and terrains by adapting to different conditions. Applications for snake robots include:

- Industrial inspection
- Scouting & recon
- Urban search and rescue
- Medicine
- Archeology
- Painting & coating

To highlight one practical example, a snake robot was used in May 2013 at the closed Zwentendorf Nuclear Power Plant in Austria. Inspection of confined spaces like pipes and vessels were performed which would not have been accessible with traditional equipment [11].

The current iteration of snake robots (Figure 7) is a modular robot system consisting of any arbitrary amount of 1-DOF modules which are described in chapter 2.5.2. These modules, assembled as snake with a front gripper, can serve as a robot manipulator if needed. Furthermore, the modules can be connected with connector links to form a regular robot arm.

An evolution of the SEA modules is the newly developed X5 series, which are described in chapter 2.5.3. They were designed to create modular robotic manipulators depending on the application domain.

Both modules are controlled via MATLAB[6] and the HEBI-API. The X5 series and the HEBI-API are produced and maintained by Hebi-Robotics[7].

Figure 7 - Modular SEA Snake

2.5.1 Series Elastic Actuators

Series Elastic Actuators are equipped with an internal passive mechanical spring initially proposed in Series Elastic Actuators by Pratt et al. [12]. During actuator control, the spring deformation can be measured and converted into applied torques. A unique design was developed for the snake modules to allow for general or impedance control. This is important in areas in which robots are interacting with the world by using force or impedance control. By measuring forces which are applied, new control algorithms can be designed and damage to objects or the robot can be prevented. The spring in the snake modules consists of a rubber elastomer which allows for elasticity by torsional shearing. The elastomer is fixated to two rigid plates as shown in Figure 8. Torques exerted by the joint can also be measured by the motor current, but spring measurements showed to be more accurate. Further details about the SEA snake modules can be found in [13]. The importance of these modules is shown in chapter 5 for the dynamic parameter identification.

Figure 8 - Series Elastic Spring with Cross-Section (right)

[6] http://www.mathworks.com/

[7] http://hebirobotics.com/

2.5.2 SEA Snake Module

The SEA Snake modules, as shown in Figure 9, are 1-DOF modules which are an improvement over the previously used Unified Snake modules. Especially the use of a series elastic element for torque sensing, as stated in chapter 2.5.1, allows for new research and operation domains. Table 1 lists the specifications of the SEA Snake modules. Further information about the SEA Snake and its modules can be found in [14].

Figure 9 - SEA Snake modules

Dimensions	Diameter	5.1 cm
	Length	6.4 cm
Mass	Module	205 g
Actuation	Max Torque	7 Nm
	Current resting	40 mA
	Current max	600 mA
Power	Voltage	24-48 V
Communication	Ethernet	100 Mbps
Sensing	Angular Position and Velocity Output Torque 3-Axis Accelerometer 3-Axis Gyro Temperature Voltage Current	

Table 1 - SEA Snake Module Specifications

2.5.3 X5-Series Industrial Smart Actuator

The X5 series, shown in Figure 10, is geared towards researchers, engineers, and industrial integrators. Its modular and lightweight design allows to quickly create custom robot configurations. Table 2 lists the specifications of X5 modules

As with the SEA modules, they are equipped with low-cost but reliable Inertial Measurement Units (IMU). The input commands also allow for position, velocity and torque control. The series elastic actuator is driven by a brushless motor which allows for continues output rotation, unlike the limited range of motion of the SEA modules. To control the actuators, APIs for MATLAB, ROS and C/C++ are supported. Further details can be found in [15].

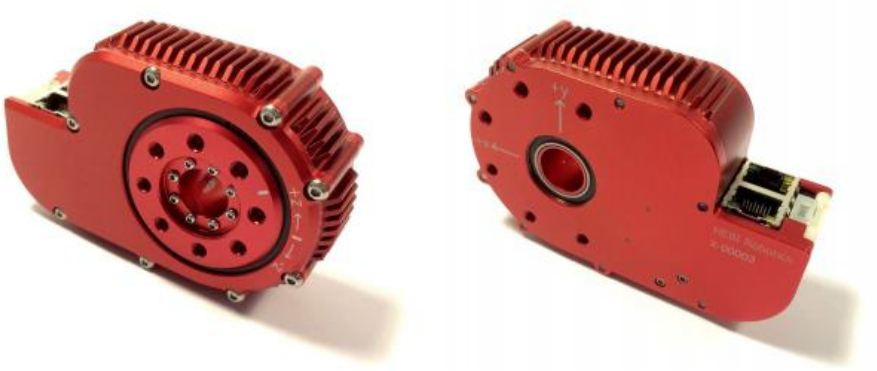

Figure 10 - X5 Series

Dimensions	Size	31 mm x 110 mm x 73 mm
	Hollow Bore	15 mm
Mass	Module	306 g
Actuation	Max Speed	14 RPM
	Max Torque	13 Nm
	Current resting	0.8 A
	Current max	2.4 A
Power	Voltage	18-50 V
Communication	Ethernet	100 Mbps
Sensing	Angular Position and Velocity Output Torque 3-Axis Accelerometer 3-Axis Gyro Temperature Voltage Current	

Table 2 - X5 Series Module Specifications

The accelerometer and gyroscope alignment are shown in Figure 11. The *x*- and *y*-axis are indicated by the arrows. The *z*-axis is pointing straight out of the picture. For the link length estimation, the location of the IMU in the module must be known. The blue square indicates the position of the sensors.

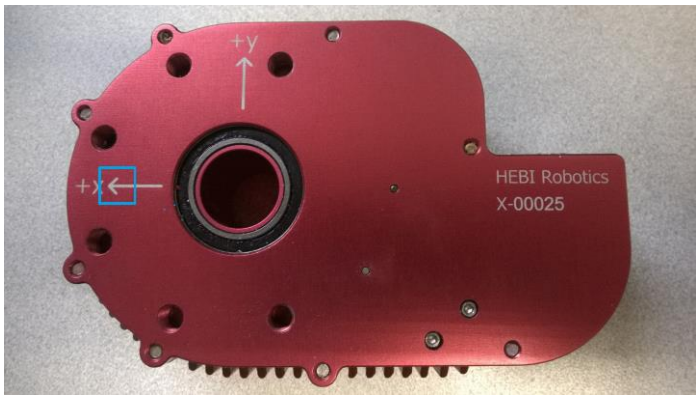

Figure 11 - X5 Sensor Axis and Position

2.5.3.1 Control Modes

As stated in chapter 2.5.3, the HEBI actuators can be controlled by supplying position, velocity, and torque commands. PID control loops are used to combine these three inputs. The control modes use different control loop sequences depending on which input signal should be favored. Every PID controller for each input signal (position, velocity, torque) has

their own PID parameters. This means, that the proportional, integral and derivative part for each of the three controllers has to be tuned. There are two prominent control modes right now, Control Mode Strategy (CMS) 3 for position control and Control Mode Strategy 4 for torque control [16].

The CMS 3 has no outer torque loop and is used for position control. As displayed in Figure 12, position, velocity and torque PID are summed to generate the motor signal.

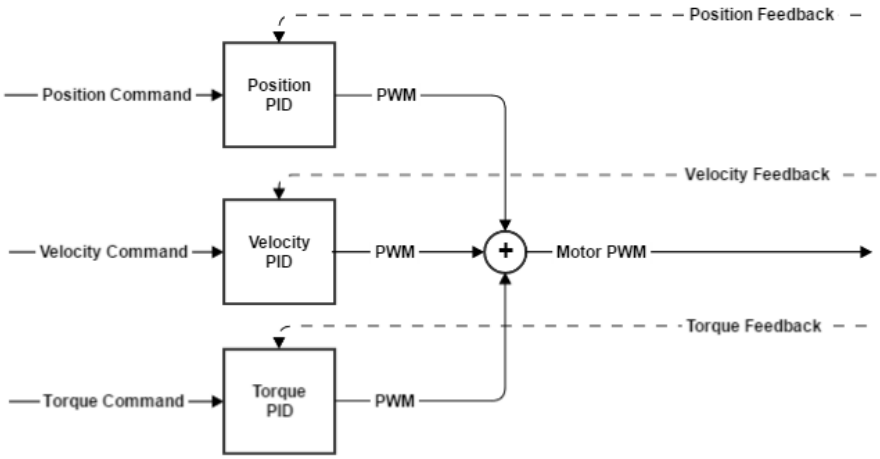

Figure 12 - Control Mode 3 for Position Control

For torque control this mode is used. A feed-forward torque signal is summed with the position PID output to produce a torque signal with is then passed to the inner torque PID controller. That output is then summed with the velocity PID signal to generate the motor PWM commands as shown in Figure 13.

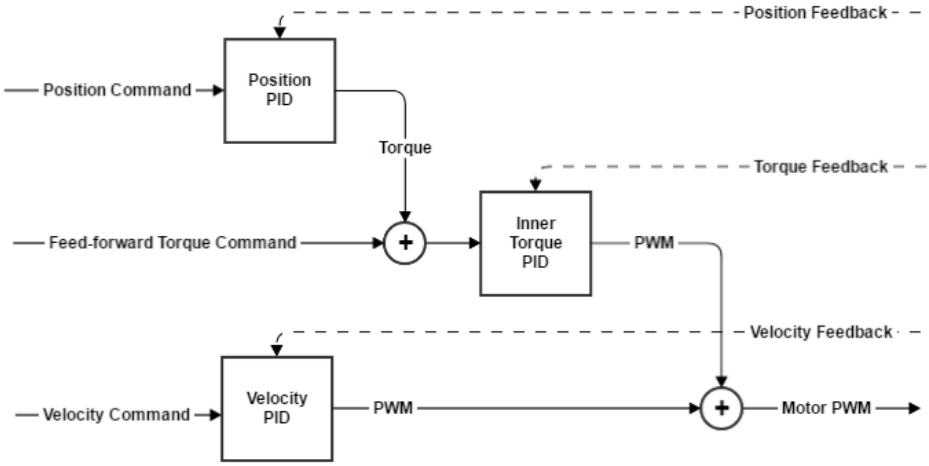

Figure 13 - Control Mode 4 for Torque Control

3. Related Work

In robotics, a vast amount of different models exists, which are all needed for proper control of a robot. Actuator models, sensor models, kinematic models, dynamic models, and others. The field of system identification deals with the problem of identifying such models from measurements. In general, these models can be grouped into two categories:

- Parametric models
- Non-parametric models

Parametric models are characterized by a few parameters, which are enough to describe the model in an accurate enough fashion. These often include manmade parts, where the properties are understood and controlled like in this thesis. These include DH parameters (chapter 2.3.1.1), and inertial parameters of rigid bodies (chapter 2.4).

Non-parametric models are used for complicated systems, where a handful of parameters isn't enough like for biological systems. Used as a stepping stone, they can help to create parametric models. As example, Bode plots can be used to analyze if a system should be handled as a second- or third-order system [17].

3.1 Kinematic Identification

While kinematic parameter identification is often not an issue discussed with professional robots in high performance environments, finding kinematics for experimental robots for use cases like error handling or self-configuration has been a topic of increased importance. Especially in recent decades a variety of detailed publications shows the importance of kinematic calibration [18] [19] [20].

Calibration routines differ widely, but having a robust and versatile procedure requires to take various points into consideration. Often there is a tradeoff made between following points [19]:

- Variety of approaches for different circumstances
- Statistically robust
- Measurement equipment is accurate, easy, inexpensive
- Calibration should require minimal human involvement

Furthermore, kinematic parameter identification can be classified into the categories discussed in the following chapters.

3.1.1 Open-loop methods

These have the robot moving freely in space without the end-effector constraint to a fixed position. As stated in [20], these methods are heavily dominated by external measuring systems [21] [22]. Cameras, or tools like laser sensors [23], track the end-effector pose. Different poses are often created by moving all robot joints. Nonlinear optimization is then used on the set of measured poses to find the desired parameters [18].

Sensor systems like IMUs can be used in combination with cameras. The resulting measurement data can be fused for better results with Kalman Filters [24] or Extended Kalman and Particle Filters [25].

Another approach is using a single IMU firmly fixated on the end-effector. In [26], kinematic calibration is performed by using IMU data and joint values. Single joint movements using Circle Point Analysis (CPA) to gather sensor data. Similar, but more sophisticated approaches are performed in [27] and [28], by using a Factored Quaternion Algorithm, a Kalman Filter and an extended Kalman Filter.

In [29], which builds on Rigid Body Load Identification for Manipulators by Atkeson et al. [26], a distributed IMU network on the robot links is used to find the kinematic parameters. Their idea improves on [26] by not using error inducing first order integration and needing lower joint speeds. Parts of this approach are used in this thesis and further details are outlined in chapter 4.

3.1.2 Closed-Loop Methods

The end-effector is firmly attached to a fixed point in space to form a closed kinematic chain. This approach doesn't need external measurement system and can rely solely on the joint angles. End-effector fixation can vary from full six-degree constraint to only one-degree of motion constraint [30] [31] [32].

3.2 Dynamic Identification

Same as with kinematic parameter identification, dynamic parameter identification has been an important research interest for several decades already. Procedures are again separated in open-loop and closed-loop approaches. In 1985, Atkeson et al. proposed the identification of inertial parameters [33]. His approach, also called Inverse Dynamic Identification Model using Least Squares (IDIM-LS) [17], is used in this thesis and further discussed in chapter 5.

A recent and common approach is the Closed Loop Output Error (CLOE) method [34] and recent improvements which are built upon it like the Direct and Inverse Dynamic Identification Model (DIDIM) [35]. The DIDIM method minimizes a 2-norm error between real forces/torques and simulated ones. In [36] identification is proposed by calculating the energy or power of the system using a so called Power Identification Model.

4. Kinematic Parameter Identification

Most related work mentioned in chapter 3.1 makes use an external measurement system like a camera, others use closed-loop approaches. Neither of these approaches is applicable for the SEA and X5 modules used, because having such a system in place cannot be guaranteed. Several important factors are taken into account for choosing the right kinematic identification scheme.

Identification should be robust and not require much a-priori knowledge of the robot. Moreover, no physical interaction by a human operator should be required, and the scheme should be sufficiently fast for real world application.

The use of external measurement systems, like cameras, is not always applicable. Some environments might not allow for a proper camera setup. Additionally, such a system needs calibration and adds another point of failure and errors. Therefore, if possible, only proprioceptive sensors should be used. As stated in chapter 2.5.3, the sensors in the X5 modules are low-cost. Thus, some inherent noise, drift and lower accuracy is to be expected compared to expensive sensors which might even not be single-chip devices like the IMUs used in the SEA and X5 modules. As a consequence, an appropriate procedure must account for that.

Closed-loop approaches can make use of the internal sensors, but such approaches are again not always applicable depending on the circumstances. A manipulator on a remote robot might not be able to form a closed-loop. Furthermore, a closed-loop can inflict damage to the manipulator or the environment.

Approaches using an IMU located on the manipulator's end-effector showed promising results as shown in [26]. The advantage of the SEA and X5 modules is, that every module, not just the end-effector, has a dedicated IMU inside. Accordingly, an approach which makes use of the multi-sensor setup desirable, and might improve accuracy.

A fairly recent, novel, and interesting approach is the kinematic parameter identification proposed by Philip Mittendorfer et al. [29]. It uses a manipulator equipped with artificial skin modules on random places on its links. These modules are equipped with low-cost three-axis gyroscopes and accelerometers and are used to determine the manipulator topology and the kinematic parameters in an automated fashion. Because of random sensor placements, sophisticated algorithms are used to extract the parameters. Additionally, their experiments show good results with translation errors of ≤0.05 m and rotational displacement of ≤0.09 rad.

Given these promising results, their unique approach, and unfeasible approaches of related work, their kinematic identification scheme is used in this thesis. Since the SEA and X5 modules already have an IMU built in, the procedure for the parameter identification is simplified. Especially for finding the joint axis ${}^{u}j_{d}$, a simpler identification scheme is used.

4.1 Constraints

The routine requires that all the joints are revolute, and since there are no prismatic SEA or X5 modules, this doesn't affect the outcome. The motors must be able to measure their position, and if possible velocity, so that the join accelerations can be calculated by single or double differentiation.

The first link or motor has to stay stationary for the whole routine to have a reference body frame that is not moving.

Furthermore, the radial distance of a link must be over a certain threshold. A shorter link will increase the estimation errors.

4.2 Circle Point Analysis

The Circle Point Analysis (CPA) [37] is used to identify several parameters of joint *d* with respect to a base frame *u*. Each of the joints is moved one at a time, while the others remain fixated. While following a circle, the trajectory of the joint described is measured by the IMUs in the next joint at the end of the link. These special trajectories are the motion patterns described in chapter 4.3.1. Figure 14 shows the circle trajectory of a joint with its parameters.

The following two circle point vectors can be calculated by the CPA:

- ${}^{u}\mathbf{j}_{d}$: rotation joint axis unit vector
- ${}^{u}\mathbf{r}_{d}$: tangential unit vector towards the next joint

The acceleration components measured by the accelerometers in the IMU are the following:

- ${}^{u}\mathbf{g}_{d}$: gravity vector
- ${}^{u}\mathbf{a}_{tan,d}$: dynamic tangential acceleration
- ${}^{u}\mathbf{a}_{cp,d}$: dynamic centripetal acceleration

Each one is dependent on one of the joint variables $\varphi_d, \omega_d = \dot{\varphi}_d$ or $\propto_d = \dot{\omega}_d$, and the circle point vectors ${}^{u}j_d$ and ${}^{u}r_d$.

These three accelerations make up the full measured acceleration vector, shown in Equation 12:

$$ {}^{u}a_d = {}^{u}a_{tan,d} + {}^{u}a_{cp,d} + {}^{u}g_d \tag{12}$$

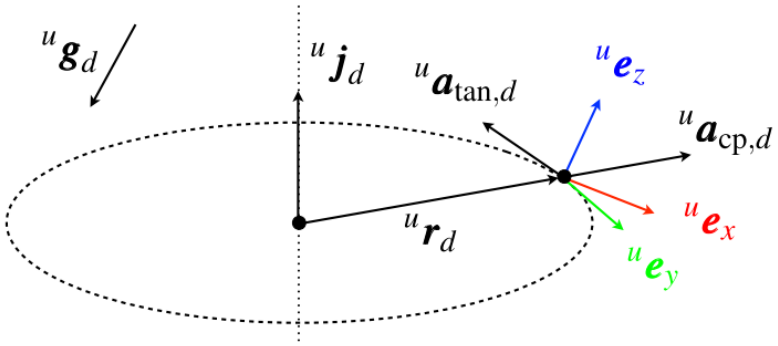

Figure 14 - Circle Point Analysis Paramters

By using the following facts, special motion patterns can be used to extract the parameters of interest:

- the dynamic accelerations $^u a_{tan,d}$ and $^u a_{cp,d}$ are orthogonal
- the dynamic components do not change their direction to the accelerometer frame *u*
- each component depends on a joint variable

4.3 Procedure

The full procedure consists of five steps and is explained further in the following chapters:

- Exploration Motions
- Joint axis unit vector estimation
- Tangential unit vector estimation
- Radial distance estimation
- Calculation of radial vector

4.3.1 Motion Patterns

The motion patterns are used to create specific trajectories which produce accelerometer and gyroscope readings which are used for the estimations.

4.3.1.1 Constant Motion

In [29], Mittendorfer et al. perform a quasi-static motion to gather gravity samples with the distributed and random placed IMUs on the manipulator links. This data is then used to compute the joint axis $^u j_d$ with a minimization algorithm. Contrary to this, the fact that the IMUs are located in the SEA and X5 modules is used. A Constant Motion (CM) with constant velocity *A* is executed, as displayed in Figure 15 and Equation 13:

$$w_{d,cm}(t) = A \tag{13}$$

The CM gathers gyroscope data, and allows to identify the rotation axis. The procedure is outlined in chapter 4.3.2.

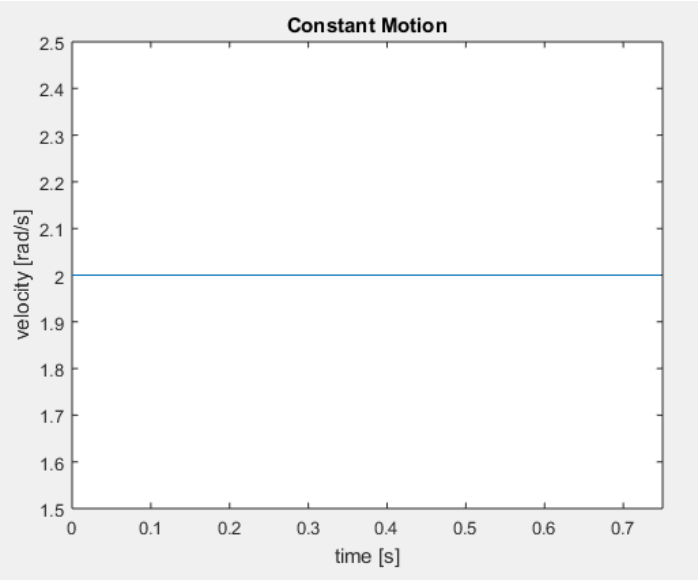

Figure 15 - Constant joint motion to determine joint axis

4.3.1.2 Windows Sine Motion

The Windowed Sine Motion (WSM) is a trajectory to maximize joint accelerations, while keeping position changes, joint velocity and jerk at a minimum. It is used to estimate the tangential acceleration vector ${}^{u}n_d$, which is explained in chapter 4.3.3. The motion for the X5 module is, as in [29], a sine with a Gaussian window function as displayed in Equation 14 and Figure 16:

$$w_{d,wsm}(t) = e^{-\frac{(t-T_m)^2}{2T_\sigma^2}}(A\sin(2\pi f t)) \tag{14}$$

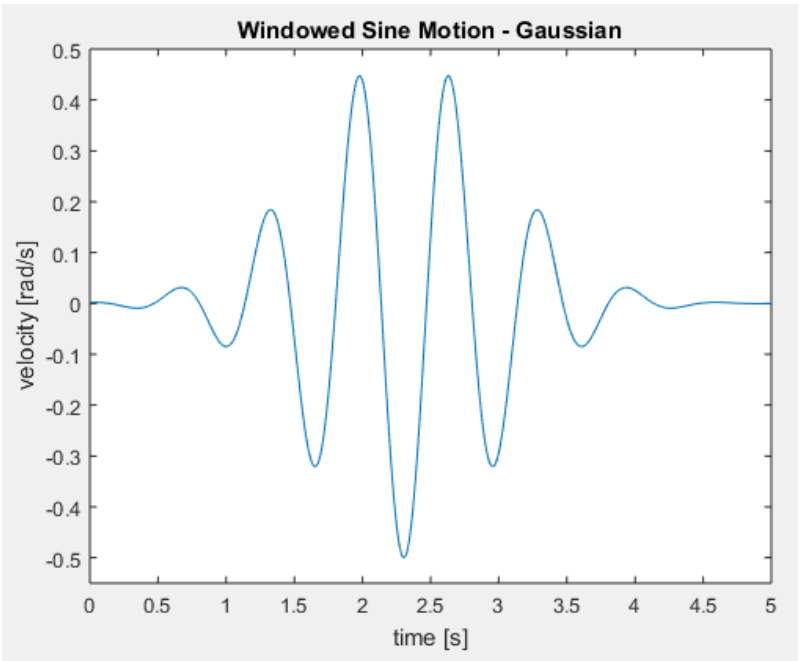

Figure 16 - Windowed Sine Motion - Gaussian Window

4.3.2 Joint Axis Unit Vector Estimation

The samples recorded during CM, and the fact that the IMUs are all fixated on the joints are used to estimate the joint axis unit vector ${}^{u}j_d$. The gyroscope readings $\omega_{cm,d}$ are used to determine the axis with respect to the IMUs coordinate frame *u*. The average over all *N* data points in $\omega_{cm,d}$ is taken, and then normalized to calculate the unit vector ($\left|{}^{u}j_d\right| = 1$):

$$ {}^{u}j_d = \frac{\frac{1}{N}\sum_{n=1}^{N}\omega_{cm,d}[n]}{\left|\frac{1}{N}\sum_{n=1}^{N}\omega_{cm,d}[n]\right|} \tag{15} $$

4.3.3 Radial Unit Vector Estimation

The dynamic acceleration samples recorded during the WSM are used to estimate the tangent unit vector ${}^{u}n_d$ ($\left|{}^{u}n_d\right| = 1$) with respect to the IMUs coordinate frame *u*.

At first, gravity values which are acting on the manipulator and sensors during the motion are calculated using Equation 16. *Rot* indicates the axis-angle rotation formula explained in chapter 2.2. The MATLAB command is *vrrotvec2mat* which takes two input arguments. The axis of rotation ${}^{u}j_d$, and the angle moved from time step zero for the sample point at time step *n*. This way, the gravity over the whole motion can be calculated by adjusting the initial gravity vector at time step zero along the whole trajectory given the angle.

$$ {}^{u}g_{,d}[n] = Rot({}^{u}j_d, \varphi_d[\mathrm{n}] - \varphi_d[0]) * {}^{u}g_d[1] \tag{16} $$

Afterwards the dynamic acceleration components can be calculated by subtracting gravity from the WSM measurements at each time step *n*:

$$ {}^{u}a_{dyn,d}[n] = {}^{u}a_d[n] - {}^{u}g_d[n] \tag{17} $$

Additionally, to prevent any interference with the singular value decomposition (SVD), the remaining mean value is subtracted to counteract an incomplete gravity subtraction:

$$ {}^{u}\tilde{a}_{dyn,d}[n] = {}^{u}a_{dyn,d}[n] - mean({}^{u}a_{dyn,d}[n]) \tag{18} $$

All data points are then assembled into a matrix ${}^{u}A_d$ for the SVD:

$$ {}^{u}A_d = [{}^{u}\tilde{a}_{dyn,d}[1], \dots, {}^{u}\tilde{a}_{dyn,d}[N]]^T \tag{19} $$

An SVD (MATLAB command *svd*) is then performed on ${}^{u}A_d$ to calculate the eigenvector Matrix ${}^{u}V_d$:

$$ [{}^{u}U_d, {}^{u}S_d, {}^{u}V_d] = svd({}^{u}A_d) \tag{20} $$

As stated by Mittendorfer et al. in [29], the fact of the orthogonality between the tangential and centripetal acceleration, and the use of a sinusoidal motion pattern, is used. The tangential unit vector ${}^{u}n_d$ is then the largest, thus first eigenvector of ${}^{u}V_d$.

$$ {}^{u}n_d = [{}^{u}V_{d,11}, {}^{u}V_{d,21}, {}^{u}V_{d,31}]^T \tag{21} $$

4.3.4 Radial Distance Estimation

To find the radial distance ${}^{u}d_d$ from joint *d* to IMU *u* at the end of the connecting link, a least squares optimization algorithm is applied. Here, the fact that at each time step *n*, the tangential acceleration is the product of the angular acceleration in the joint times the radial distance is used:

$$ {}^{u}a_{tan,d}[n] = \propto_d [n] * {}^{u}d_d \tag{22}$$

A least squares minimization is then executed to estimate ${}^{u}d_d$ by minimizing Equation 23:

$$ {}^{u}d_d = \min_{{}^{u}d_d} \sum_{n=1}^{N} ({}^{u}a_{tan,d}[n] - (\propto_d [n] * {}^{u}d_d))^2 \tag{23}$$

The raw data is sampled with the same sampling rate and doesn't have any mutual delays, so no up or down sampling is needed.

4.3.5 Radial Vector Calculation

Finally, the radial vector ${}^{u}r_d$ from the joint axis to the IMU can be computed. By taking the cross product between the plane formed by ${}^{u}n_d$ and ${}^{u}j_d$ the perpendicular vector ${}^{u}r_d$, pointing towards the joint, can be calculated:

$$ {}^{u}r_d = {}^{u}d_d * ({}^{u}n_d \times {}^{u}j_d) \tag{24}$$

5. Dynamic Parameter Identification

For the dynamic parameter identification an algorithm developed by Atkeson et al. is used [33]. This approach utilizes the torque sensor in each joint to measure the forces exerted. Their work relies on an earlier paper, where inertial parameters of an end-effector load are identified [38]. It relies on estimating the inertial parameters by direct dynamic measurements.

While there are newer methods for dynamic parameter identification, as outlined in chapter 3.2, these approaches are much more complicated compared to the method shown by Atkeson et al. in [33]. Modeling the manipulator and simulating it, or calculating a power model is not feasible for modular robotics when configurations are rapidly changing.

5.1 Constraints

For identification, a motion pattern is executed to measure torque, joint position, joint velocity, and calculate acceleration by differentiation as already stated in chapter 4.1. Additionally, angular velocity, and angular acceleration of the torque sensing coordinate frame can be supplied. It is assumed that during identification that coordinate frame is stationary. The algorithm is again limited to revolute joints.

Due to the 1-DOF measurements of the torque sensors in each joint, not all inertial parameters can be estimated. The identification results given by the procedure are sufficient to calculate torque values, which are supplied as additional input command to assist the joints in following a given trajectory.

The basis of the algorithm uses the Newton-Euler formulation for dynamics by representing the unknown inertial parameters as linear. The measured torques are a result of the joint movements (represented in a matrix expression) and the unknown inertial parameters (represented as a vector). In a similar way like already for the kinematic identification, a least squares approach is used to calculate the final parameters.

5.2 Procedure

The procedure consists of the following three steps:

- Exploration motion
- Assembling of the motion matrix
- Inertial parameter estimation via least-squares

5.3 Newton-Euler Equations Formulation

The exact derivation of the formulas can be found in Rigid Body Load Identification for Manipulators by Atkeson et al. [38]. Below, the final equations for the identification procedure are listed. Equation 25 shows the main equation:

$$\begin{bmatrix} f \\ n \end{bmatrix} = \begin{bmatrix} \ddot{p} - g & [\dot{\omega}\times] + [\omega\times][\omega\times] & 0 \\ 0 & [(g - \ddot{p})\times] & [\blacksquare\dot{\omega}] + [\omega\times][\blacksquare\omega] \end{bmatrix} \begin{bmatrix} m \\ mc_x \\ mc_y \\ mc_z \\ I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} \tag{25}$$

or in short:

$$w = A\Phi \tag{26}$$

The equation is shortened by the terms $[\omega\times]$, $[\dot{\omega}\times]$, $[\blacksquare\omega]$ and $[\blacksquare\dot{\omega}]$ for the angular velocity and angular acceleration which equates to:

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{27}$$

$$[\blacksquare\omega] = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix} \tag{28}$$

After the motion pattern is executed and the matrix A assembled the inertial parameter vector Φ_i can be estimated with least squares in matrix notation:

$$\Phi \quad = (A^T A)^{-1} A^T w \tag{29}$$

Due to the limited DOF and torque sensing, A loses rank and is not invertible. The matrix is thus inverted by using the pseudo inverse.

With the estimated vector Φ, torques for new trajectories can now be calculated by supplying joint velocities and acceleration into A and solving Equation 26.

6. Experiments and Results

The experiments are run on X5 modules for a 1-DOF arm. The goal of the kinematic identification in chapter 6.2 is to find accurate link length. During the link length estimation also the joint frame is identified and the full kinematic forward model could be calculated. This is omitted in this thesis due to only 1-DOF arms. The link length estimation accuracy is evaluated over four different link lengths.

The dynamic identification in chapter 6.3 is performed on one link length. The parameter vector Φ will be estimated to calculate supporting torques for certain trajectories. It is then assessed if supplying the extra torque command improves the accuracy of the robot trajectory given certain PID parameters.

Furthermore, the identification schemes are tested with the links in a horizontal plane (parallel to the ground), and vertically (against gravity). This will show if gravity results in any impact on the procedures.

The code for the both the kinematic and dynamic identification experiments on the manipulators is written in MATLAB. The HEBI-API, written for MATLAB, is used to communicate with the modules and access the sensor data. The sampling and control frequency is set to 250 Hz which showed good results in terms of data points for the least-square approaches.

The range of motion for the motion patterns should be limited as much as possible. It is not always guaranteed that a robot manipulator is in a safe place with no obstacles in its near vicinity. Thus, having a small range of motion guarantees better safety for the robot manipulator, the surrounding, and humans.

6.1 Experiment Setup

Figure 17 shows the 1-DOF robot arm at link length 484 mm for vertical experiments. The other three link lengths are achieved by unscrewing the module attachment and sliding it back. The length is measured manually with a tape. The robot base is clamped to the bottom. The horizontal robot setup is shown in Figure 18.

All three modules used are from the X5 series, but the one located at the end of link is from a newer iteration. At the time of the thesis only one of those newer modules was available for experiments. The older versions don't have their IMU located at the position indicated in chapter 2.5.3 and are more prone to vibration noise. Furthermore, it must be noted that each module showed slightly different actuation behavior and reacted differently to PID settings. Inconsistencies in results could be a result from that.

This setup has the *z*-axis as rotation axis, and the *x*-axis as radial vector.

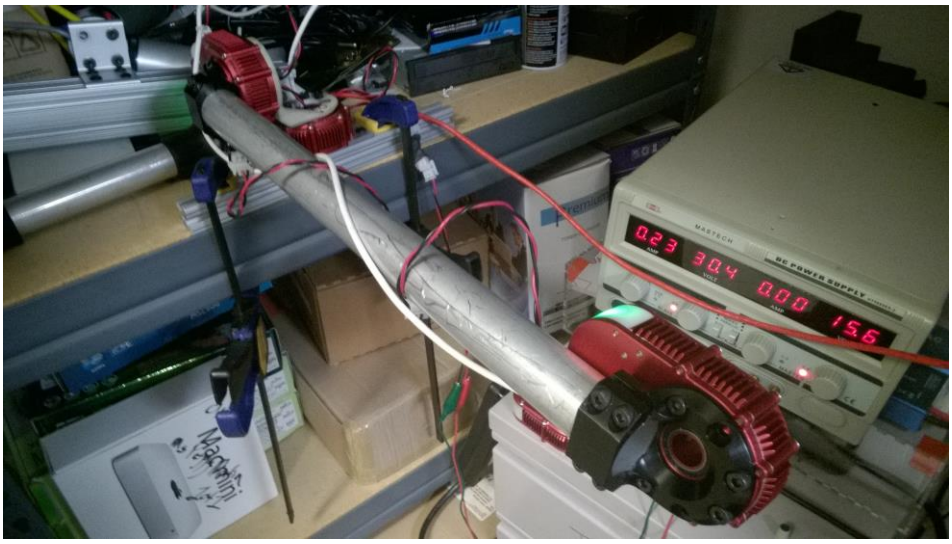

Figure 17 - Vertical Experiment Setup

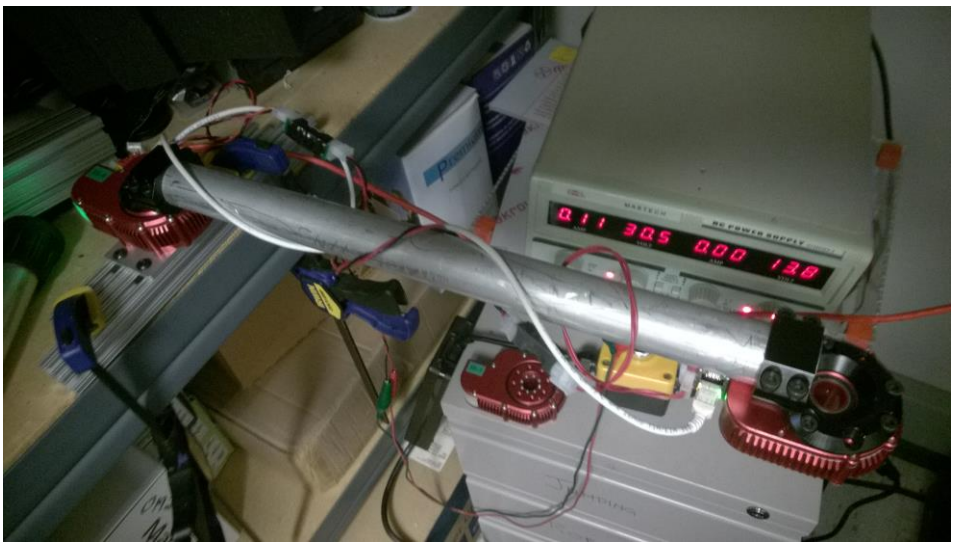

Figure 18 - Horizontal Experiment Setup

6.2 Kinematics Identification

In the following sections, the experiments for the kinematic identification algorithm are run on the X5 modules. Also problems with sensor data, and solutions are explained, and a conclusion of the algorithm is drawn.

6.2.1 Motions Patterns

To identify the right motion patterns, and especially its parameters, it is crucial for the algorithm to produce accurate results. In the following two chapters the motion details, and how to handle the sampled data is outlined.

6.2.1.1 Constant Motion

The purpose of the CM pattern is to excite the gyroscopes and identify the joint axis through it. Figure 19 shows the gyroscope signal at rest of one module. As can be seen, each axis has an inherent offset which creates errors in measurements. The offset $\omega_{offset,i}$ (i denoting the three axis x, y, and z) is calculated by sampling gyroscope values for two seconds and calculating the mean value of all data points for each of the three axis:

$$\omega_{offset,i} = \frac{1}{N}\sum_{N}^{1} \omega_{rest,i}[n] \tag{30}$$

Any measurement done for the gyroscope is now subtracted by the offset values before being used for further computation. The readings are now centered near the zero value as indicated in Figure 20.

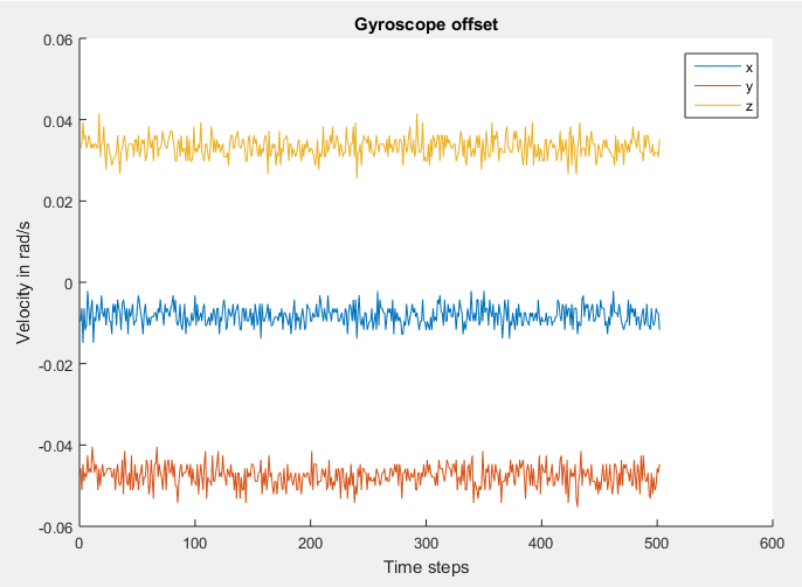

Figure 19 - Gyroscope Offset

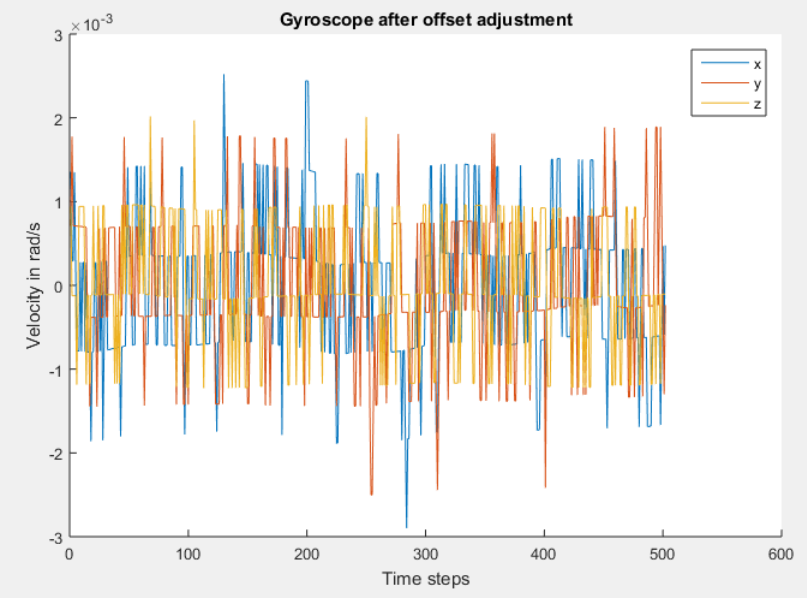

Figure 20 - Gyroscope readings after offset subtraction

For experiments the joint is positioned at a 0.5 rad offset to the zero position, and then a constant velocity of 1 rad/s is set as velocity command for the actuators. To swipe out roughly equal areas, and thus minimize motions on either side of the zero position, a motion duration of 0.75 s is used. As shown in Figure 21, the *z*-axis of the gyroscopes shows a movement in this example. The gyroscope offset is subtracted from the data points of the data samples and then the rotational joint axis is estimated as explained in chapter 4.3.1.1.

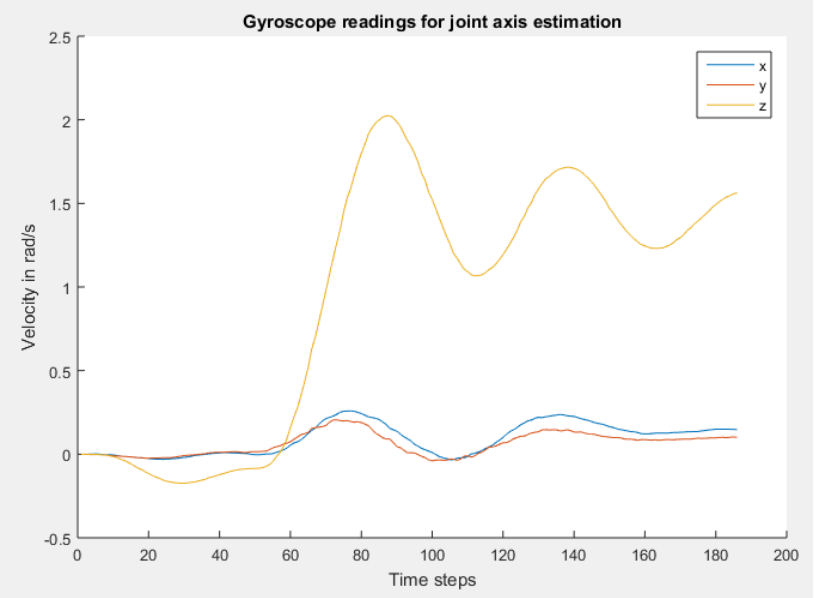

Figure 21 - Gyroscope Readings during CM for Joint Axis Estimation

Calculating the joint axis as outlined in chapter 4.3.2, with the values from Figure 21, following joint axis unit vector is estimate:

$$^{u}j_d = \begin{bmatrix} 0.0987 \\ 0.0668 \\ 0.9929 \end{bmatrix}$$

This lines up with system setup, and the statement made in chapter 6.1 that the *z*-axis is the rotation axis.

The velocity and time values used in the experiment can be chosen smaller to minimize velocity and path traveled by the link.

6.2.1.2 Windowed Sine Motion

The WSM is using the accelerometers to correlate the accelerometers with the link length as explained in chapter 4.3.1.2. Values used for the function are listed in Table 3.

f [Hz]	2
T_m [s]	2.8
T_σ [s]	1
t [s]	0: t_s:15

Table 3 – WSM Parameters for the Kinematic Identification Algorithm

Where $t_s = \frac{1}{250}$, being the time step for each input signal for the modules. The above values were identified by trial and error and showed the best results on different manipulator configurations. The amplitude value *A* has to be determined separately as explained below.

Furthermore, Figure 22 shows the resulting acceleration data unfiltered and filtered. Due to the differentiation of the velocity data to calculate the acceleration, a lot of noise is introduced. To smooth the signal, a Gaussian filter of size 11 and sigma 2.83 is used. Unfiltered acceleration produces link length estimations which are too short.

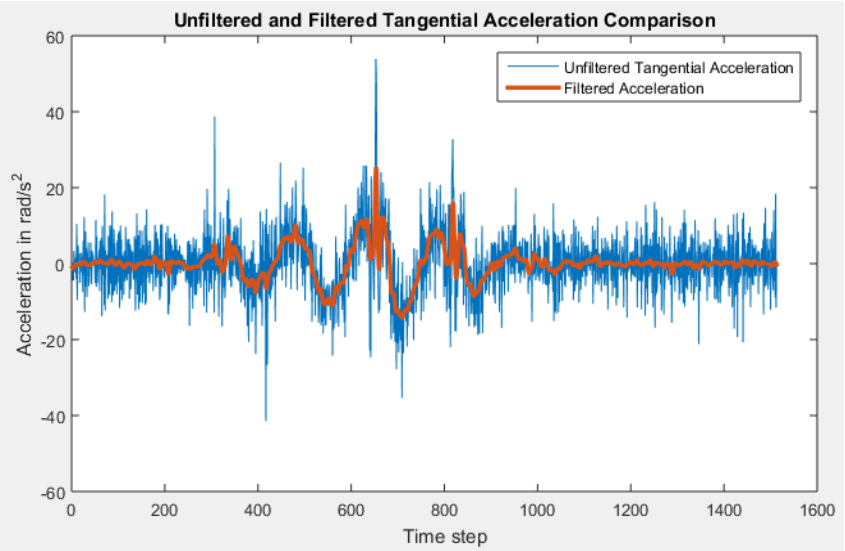

Figure 22 - WSM Acceleration Data Unfiltered and Filtered

Calculating the radial unit vector as outlined in chapter 4.3.3, with the filtered values from Figure 22, following vector is estimated:

$$^{u}n_d = \begin{bmatrix} 0.9811 \\ 0.1933 \\ 0.0017 \end{bmatrix}$$

Same as with the joint unit vector, this result aligns with the statement made in chapter 6.1 that the radial unit vector aligns with the *x*-axis of the IMUs.

Four different lengths will be evaluated with the modules. These are 484 mm, 350 mm, 245 mm, and 171 mm. Test trials showed that for each length a different amplitude resulted in an optimal result. To identify the right value, a range of different amplitudes is used on the manipulator.

6.2.1.2.1 Horizontal Amplitude Estimation

Figure 23 displays an amplitude identification run on link length 484 mm. Amplitudes from 0.02 rad up to 0.28 rad in 0.02 rad increments are used. As can be seen, 0.9 and 0.1 showed the best accuracy, coming closest to the ground truth of 484 mm.

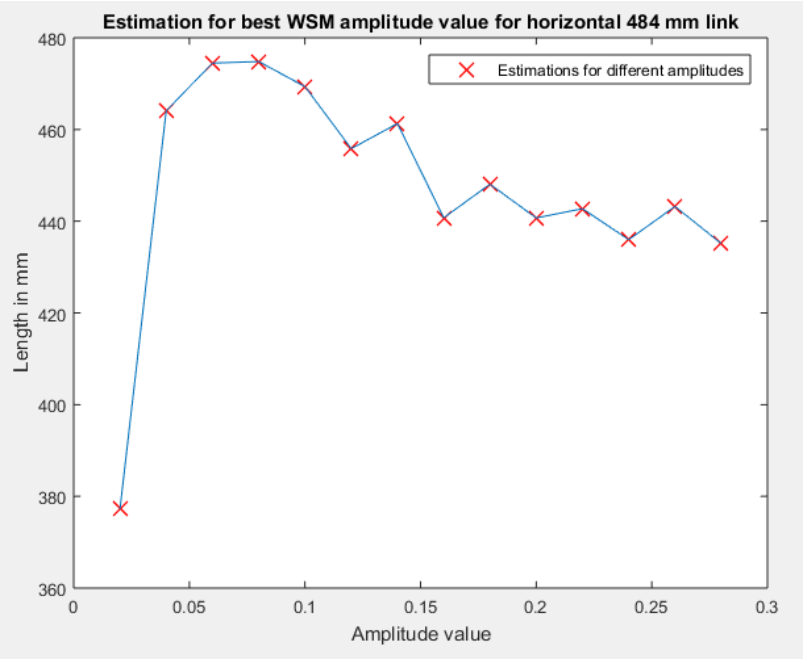

Figure 23 - Amplitude Identification for Horizontal Link Length 484 mm

Figure 24, Figure 25, Figure 26 display the same experiments for the remaining three link length. The three graphs suggest that the shorter the link, the higher the amplitude must be. This can be explained by Equation 22 from chapter 4.3.4. The identification algorithms rely on correlating IMU accelerations with joint movements. Due to the fact that smaller link lengths produce smaller tangential accelerations, higher joint velocities are needed to produce these. Therefore, higher amplitudes show better results in test trials. Contrary to that statement, Figure 26 actually shows a different behavior. The causes for the unexpected results are outlined below.

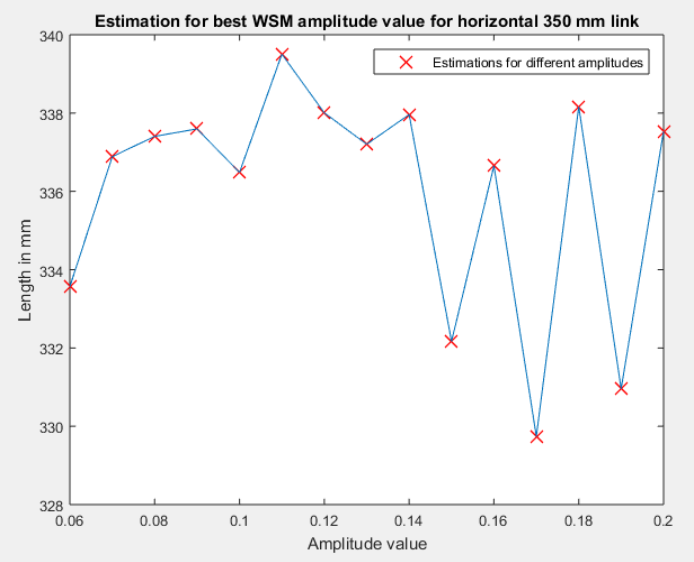

Figure 24 - Amplitude Identification for Horizontal Link Length 350 mm

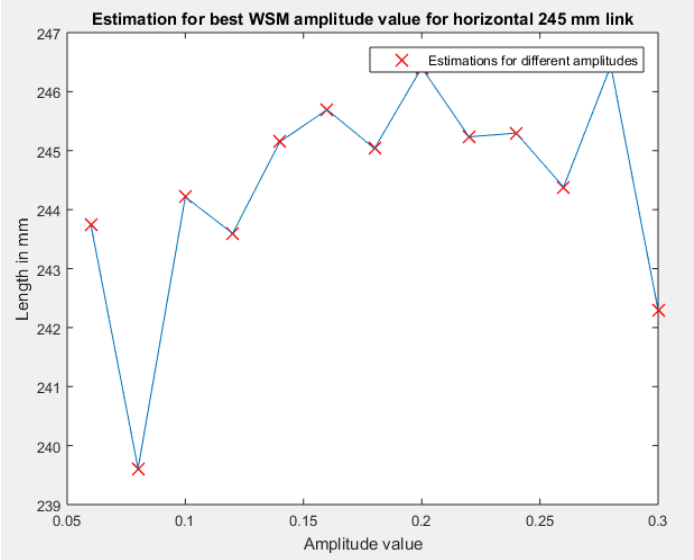

Figure 25 - Amplitude Identification for Horizontal Link Length 245 mm

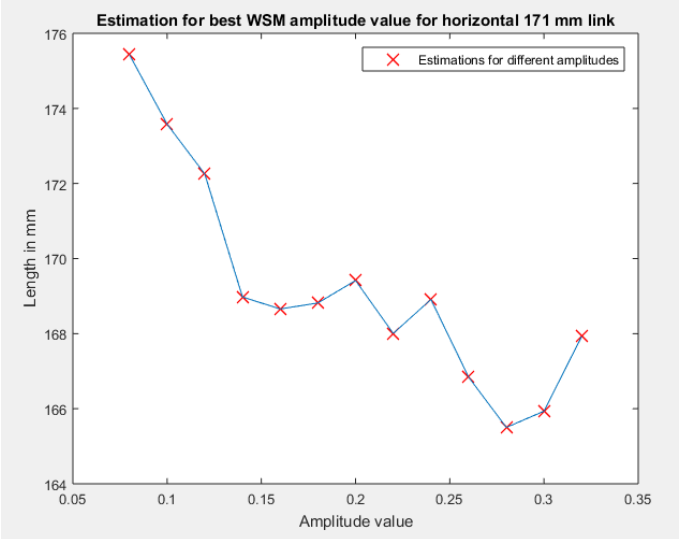

Figure 26 - Amplitude Identification for Horizontal Link Length 171 mm

For link length 171, as shown in Figure 26, overestimation can be seen for very small amplitudes too. This contradicts the statement that shorter link lengths need a higher amplitude. However, as stated in the constraints in chapter 4.1, shorter links result in higher estimation errors. This is, as stated, a consequence that resulting tangential accelerations are too small. Coupled with added noise on top of the acceleration signal, estimations show bigger errors.

Additionally, it can be seen that that too high amplitudes also cause overestimation. A reason for this is reaching maximum possible actuator velocities. Higher velocities introduce more actuator jerking and noise in sensor data. To show this effect, a WSM on a horizontal aligned link with length 484 mm and amplitude 0.08 and 0.3 is compared given its position, angular velocity, and angular acceleration. In Figure 27 the 0.08 WSM shows a smooth position trajectory and a fairly smooth angular velocity and acceleration. When examining Figure 28, it can be seen that the velocity is jerking and cutting off at the top, reaching maximum joint velocity. This is reflected in the position and angular acceleration graph showing abnormal behavior compared to Figure 27.

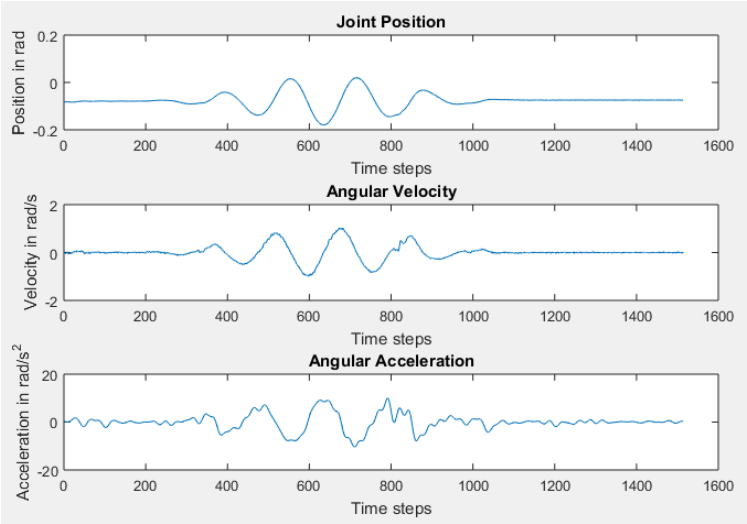

Figure 27 - WSM with amplitude 0.08 on Horizontal Joint with Length 484 mm

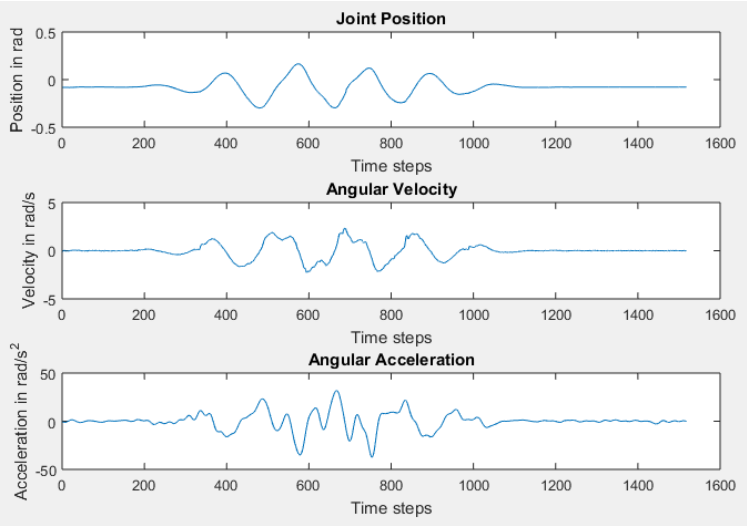

Figure 28 - WSM with amplitude 0.3 on Vertical Joint with Length 484 mm

6.2.1.2.2 Vertical Amplitude Estimation

As with the horizontal, optimal amplitudes for the WSM are evaluated on all four link lengths in vertical position to see if the pull of gravity is having an impact on the movements and algorithms. Comparing Figure 29, Figure 30, Figure 31, and Figure 32 with the Figures in the horizontal amplitude estimation chapter 6.2.1.2.1, it is apparent a different pattern is emerging.

While Figure 29 with link length 484 mm shows an expected amplitude pattern, the other three smaller lengths show a decreasing amplitude requirement to reach the desired length estimation. This is opposite to what was established for the horizontal amplitude estimation. Especially amplitudes of below 0.06 produce a very small trajectory, velocities, and accelerations, and can't possibly produce reliable estimates.

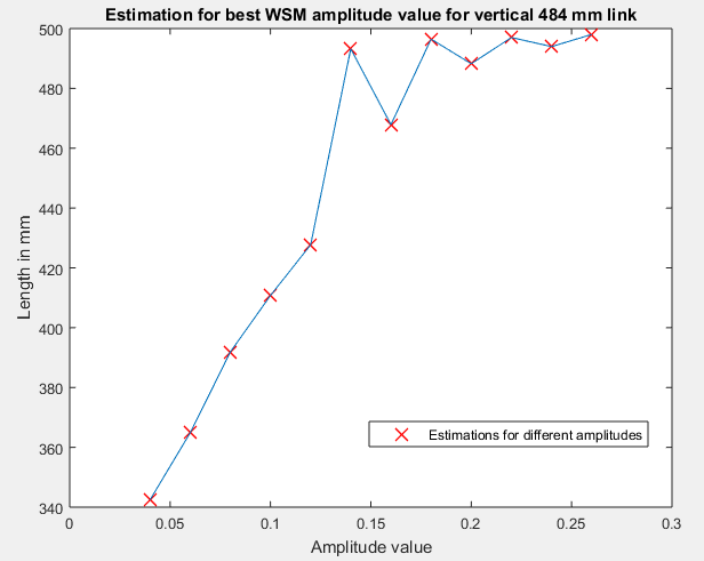

Figure 29 - Amplitude Identification for Vertical Link Length 484 mm

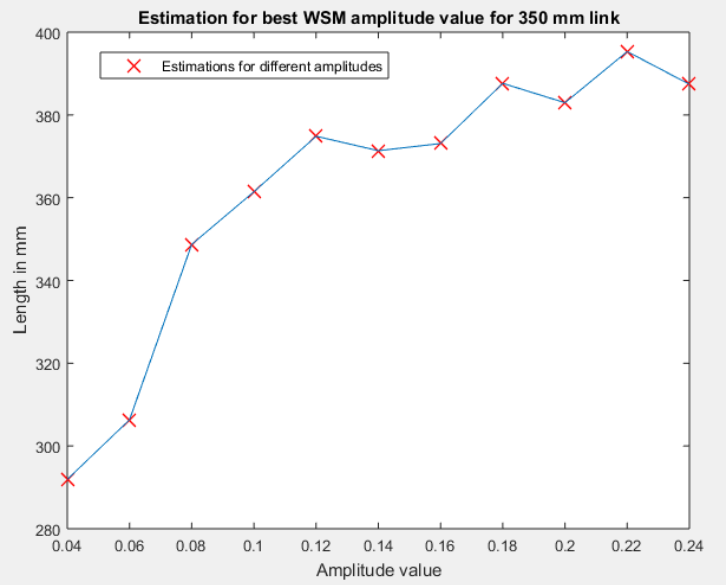

Figure 30 - Amplitude Identification for Vertical Link Length 350 mm

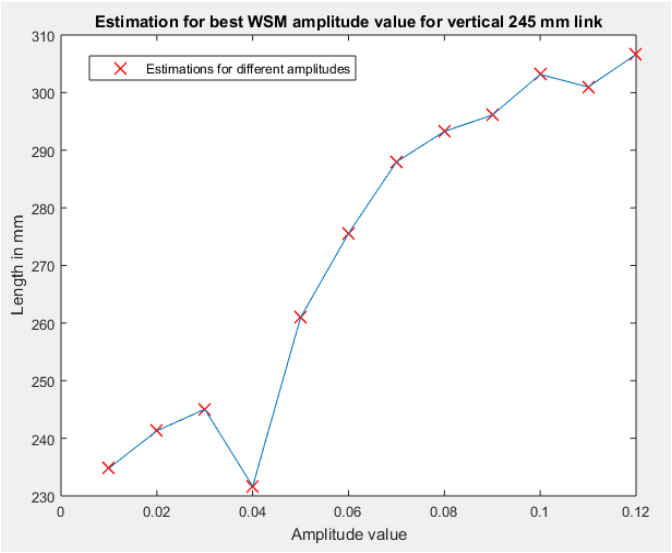

Figure 31 - Amplitude Identification for Vertical Link Length 245 mm

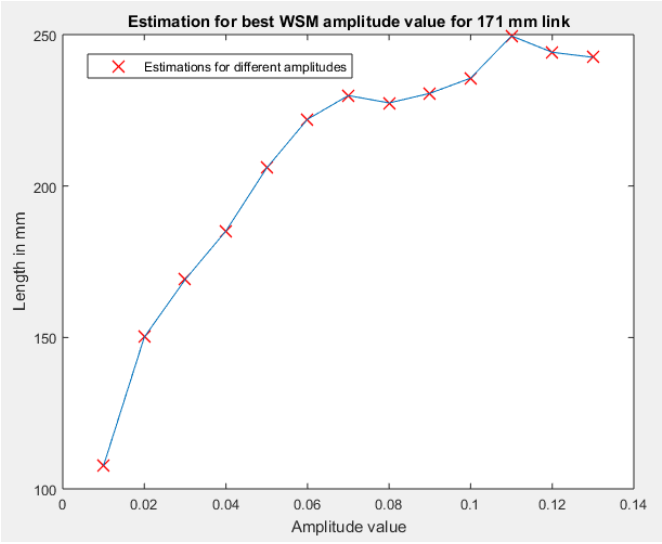

Figure 32 - Amplitude Identification for Vertical Link Length 171 mm

To acquire some insight in the unexpected behavior, the position-velocity-acceleration graph is of use again. Figure 33 shows the plot for a link length of 484 mm and an amplitude of 1.3. While the estimation shows correct results, the graph is different compared to the same horizontal link length in Figure 27. It is obvious that the actuator is having troubles performing the WSM. In Figure 27 the graph for link length 245 mm and amplitude 0.03 is displayed. The positional range of the trajectory is so small, that compared to other WSM graphs, the accelerations are very low. Looking at the graph, the estimation results seem rather like a fluke. Looking at Figure 28 with the same link length but an amplitude of 0.2, it can be seen that velocity is capped again and thus not producing proper results. In comparison, for the horizontal estimation of length 245 mm, an amplitude of 0.2 was required.

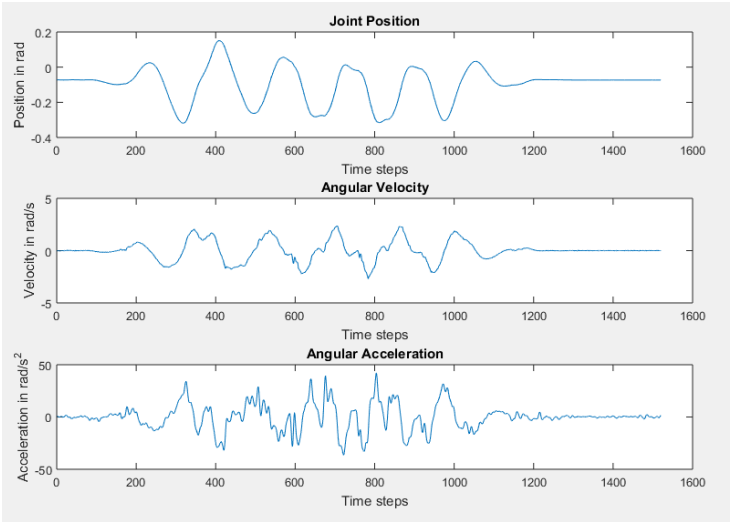

Figure 33 - WSM with amplitude 0.13 on Vertical Joint with Length 484 mm

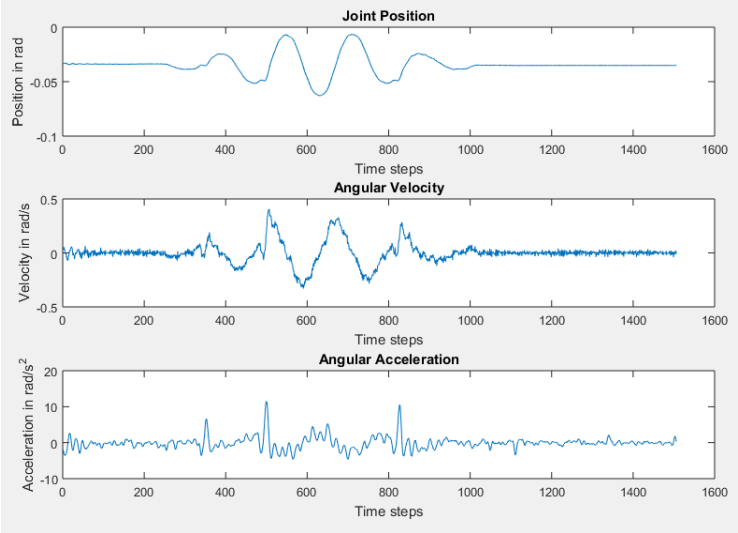

Figure 34 - WSM with amplitude 0.03 on Vertical Joint with Length 245 mm

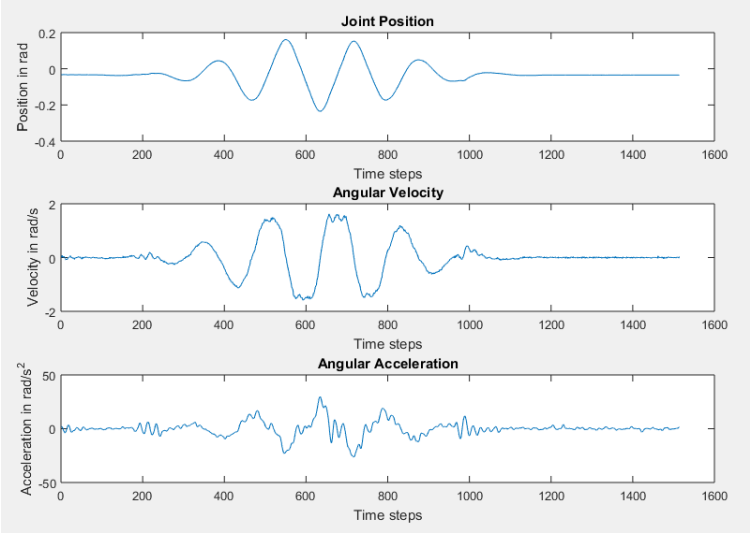

Figure 35 - WSM with amplitude 0.2 on Vertical Joint with Length 245 mm

6.2.2 Horizontal Estimation Results

Figure 36, Figure 37, Figure 38, and Figure 39 show thirteen experiment runs for each Vertical link lengths. In Table 4, the mean value, mean error in %, and mean error in mm are listed.

	484 mm	350 mm	245 mm	171 mm
Amplitude	0.09	0.11	0.2	0.2
Mean [mm]	479.6 mm	335.5 mm	245.5 mm	167 mm
Mean Error [%]	0.9 %	4.1 %	0.23 %	2.38 %
Mean Error [mm]	4.35 mm	14.5 mm	0.58 mm	4.1 mm

Table 4 - Horizontal Link Estimation Results

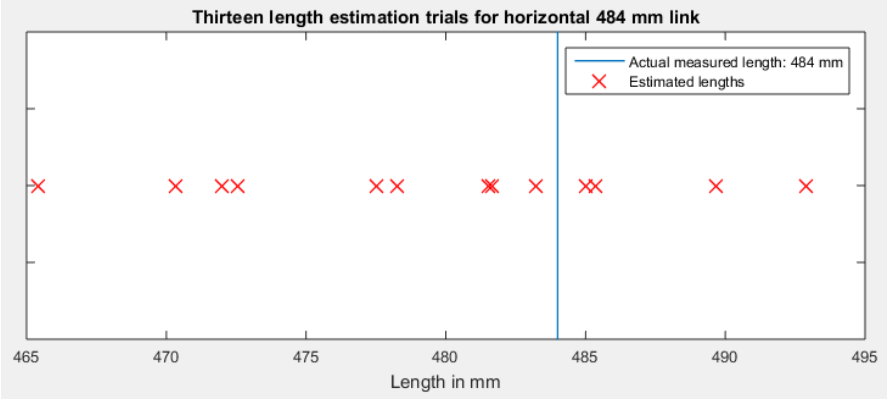

Figure 36 - Thirteen Trials with Amplitude 0.09 on Horizontal 484 mm Link

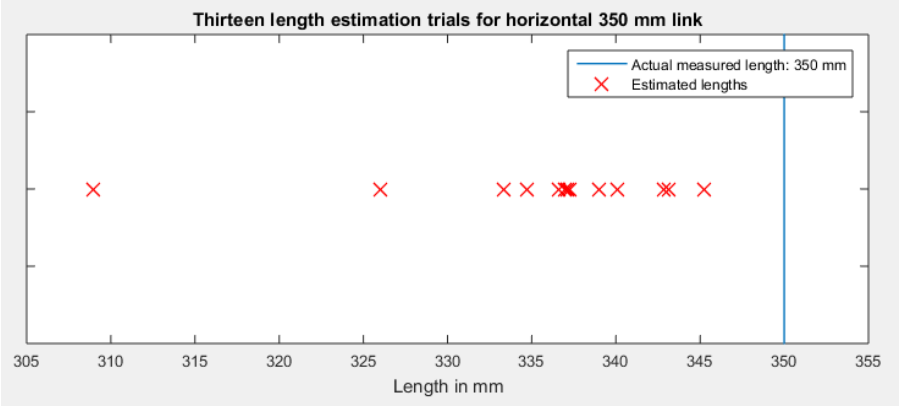

Figure 37 - Thirteen Trials with Amplitude 0.11 on Horizontal 350 mm Link

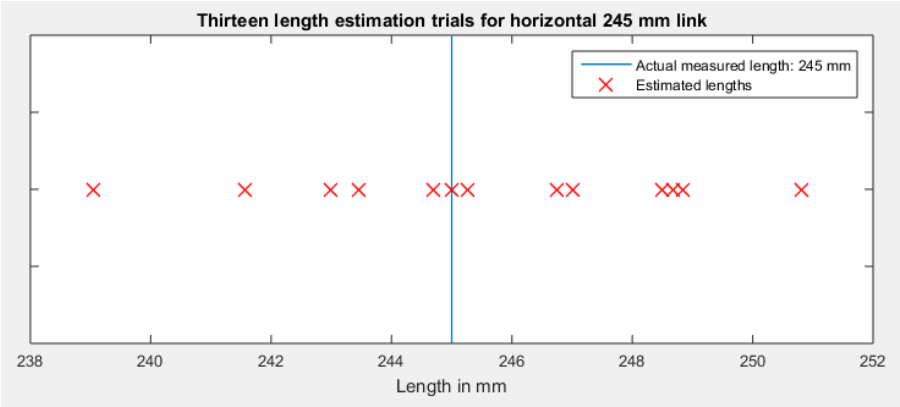

Figure 38 - Thirteen Trials with Amplitude 0.2 on Horizontal 245 mm Link

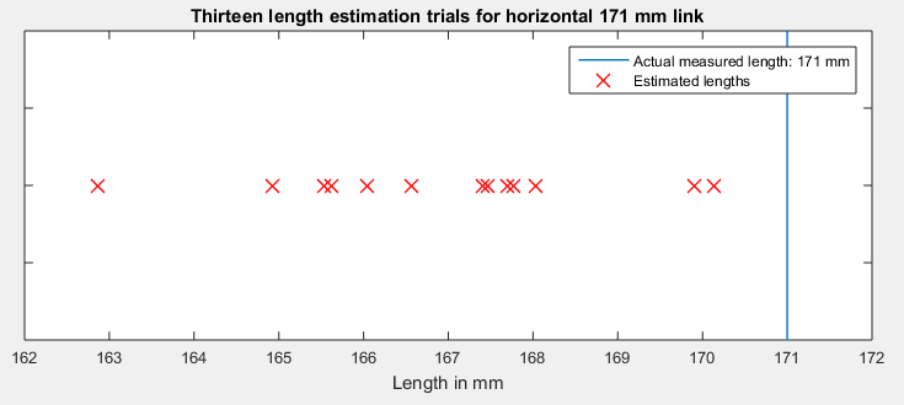

Figure 39 - Thirteen Trials with Amplitude 0.2 on Horizontal 171 mm Link

6.2.3 Vertical Estimation Results

Figure 40, Figure 41, Figure 42, and Figure 43 show thirteen experiment runs for each vertical link length. In Table 5, the mean value, mean error in %, and mean error in mm.

	484 mm	350 mm	245 mm	171 mm
Amplitude	0.12	0.08	0.03	0.04
Mean [mm]	482.3 mm	360 mm	264.7 mm	188 mm
Mean Error [%]	0.6 %	3 %	8.1%	10 %
Mean Error [mm]	1.7 mm	10.6 mm	19.7 mm	17 mm

Table 5 - Vertical Link Estimation Results

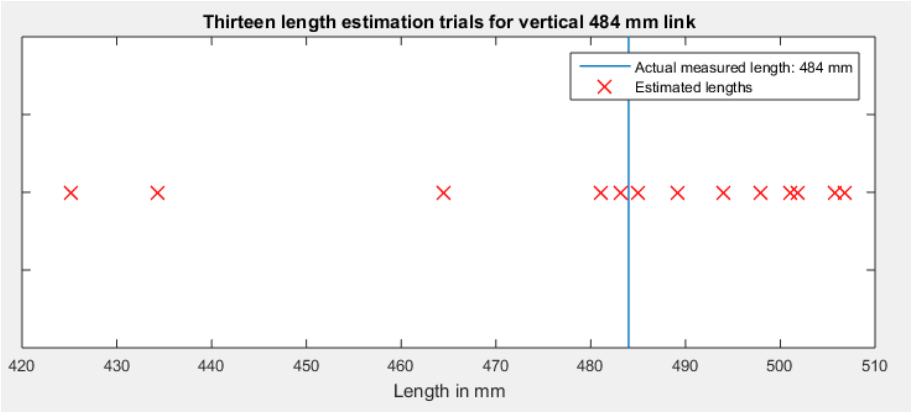

Figure 40 - Thirteen Trials with Amplitude 0.12 on Vertical 484 mm Link

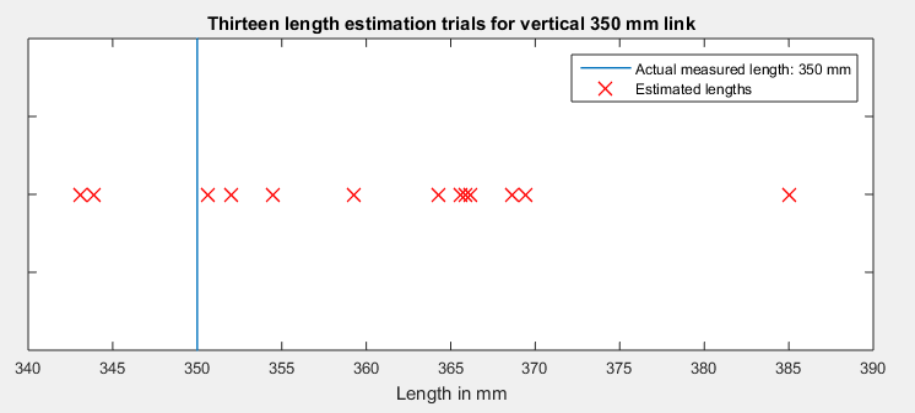

Figure 41 - Thirteen Trials with Amplitude 0.08 on Vertical 350 mm Link

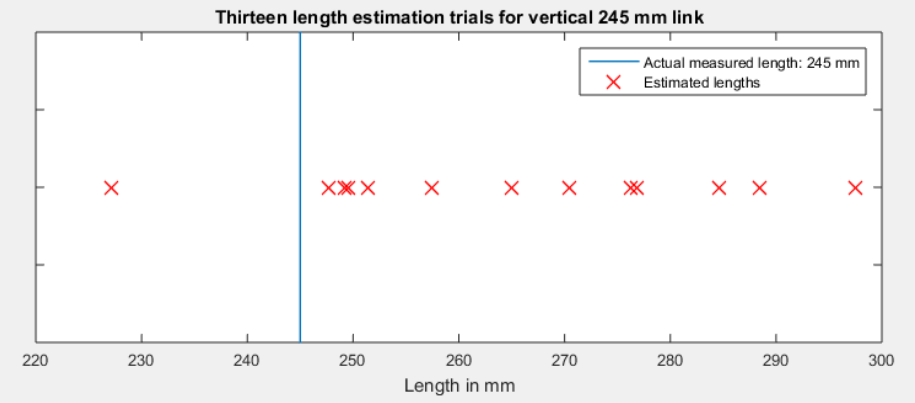

Figure 42 - Thirteen Trials with Amplitude 0.03 on Vertical 245 mm Link

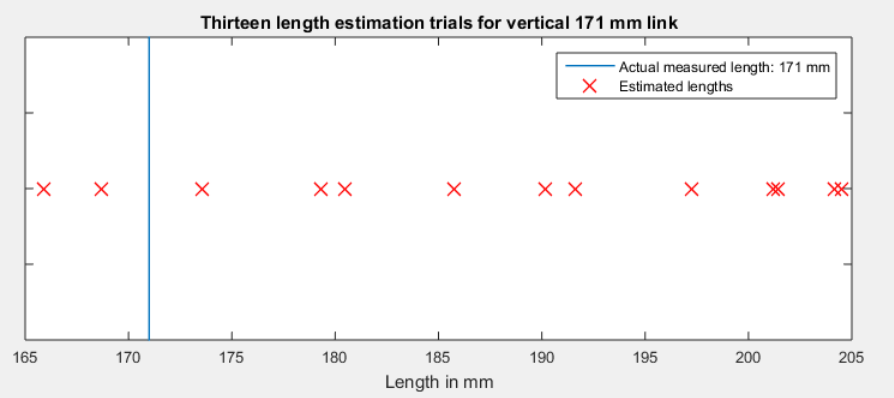

Figure 43 - Thirteen Trials with Amplitude 0.04 on Vertical 171 mm Link

6.2.4 Conclusion

Looking at the experimental results from the previous chapters a few conclusions can be drawn. While the horizontal link estimation showed good results for all four links, the vertical link estimation showed unexpected behavior.

The mean error for the horizontal links was under 5 %. Furthermore, single trials varied only by a few millimeters for all four links, making the procedure robust.

On the other hand, the vertical link estimation did not deliver the expected results. During the experiments the robot setup showed more jitters during the WSM. This could be due to the more fragile setup as shown in chapter 6.1, Figure 17. Also since the actuators are working against gravity, they are showing a worse performance.

6.3 Dynamics Identification

The dynamic identification is only performed on link length of 484 mm since we want to have a big mass and big inertia values. Supplying a torque command in such settings shows a bigger impact and verifies the viability of the routine.

6.3.1 Motion Pattern

Different amplitudes for the dynamic identification motion betters again showed different results depending on horizontal or vertical link alignment. While there is no desired goal, like with the link length estimation where the reference length is known, resulting graphs and least-squares error (LSE) between calculated and measured torque can be compared. The whole identification procedure is run, the Φ-vector estimated, and then with the estimated inertial parameters, a theoretical torque given the estimated parameters is calculated. This calculated torque input can be compared with the actual torque which has been measured on the actuators during the motion.

Torque assistance is tested on a WSM with parameters outlined in Table 6. Not the whole motion is used for the procedure. To produce high accelerations, the steep part is cut out and repeated to produce a trajectory as shown in Figure 44.

Additionally, same as in chapter 6.2.1.2, the angular acceleration data acquired by differentiation must be filtered with the Gaussian filter for optimal results.

A	0.2
f [Hz]	1
T_m [s]	2.5
T_σ [s]	0.1
t [s]	0:t_s:15

Table 6 - Motion Pattern Parameters for the Dynamic Idenfitifaction Algorithm

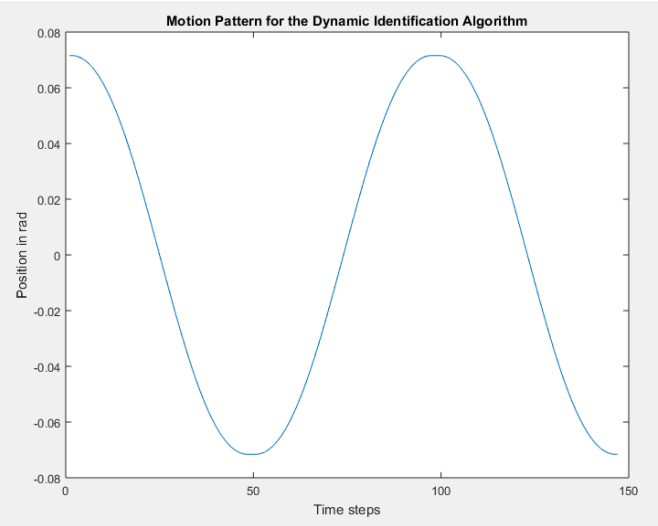

Figure 44 - Motion Pattern for the Dynamic Identification Algorithm

6.3.1.1 Vertical Amplitude Estimation

Figure 45 shows a dynamic identification run with an amplitude of 0.9 on the vertical link. The calculated torques resemble the measured torque on the actuator during the motion pattern, but they don't really line up. The LSE amounts to 107.

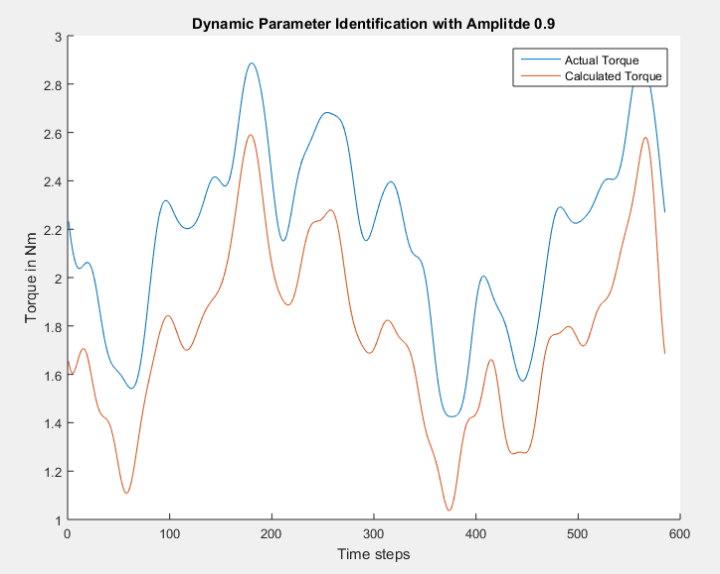

Figure 45 - Calculated and Measured Torque differences for Amplitude 0.9 on Vertical Link

Smaller amplitudes were used to see differences in torque calculations. Figure 46 uses an amplitude of 0.4 and as a result the LSE showed a lower value of 12.4, and also the overall torque requirements for the movement increased.

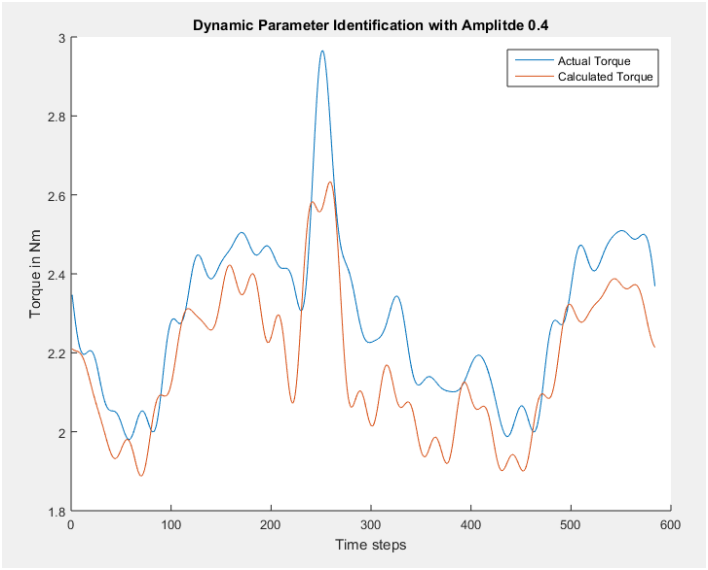

Figure 46 - Calculated and Measured Torque differences for Amplitude 0.4 on Vertical Link

Values of 0.3, 0.2, and 0.1 were tried, but as indicated in Figure 47, an amplitude of 0.2 resulted in the highest torques and an LSE of 2.0.

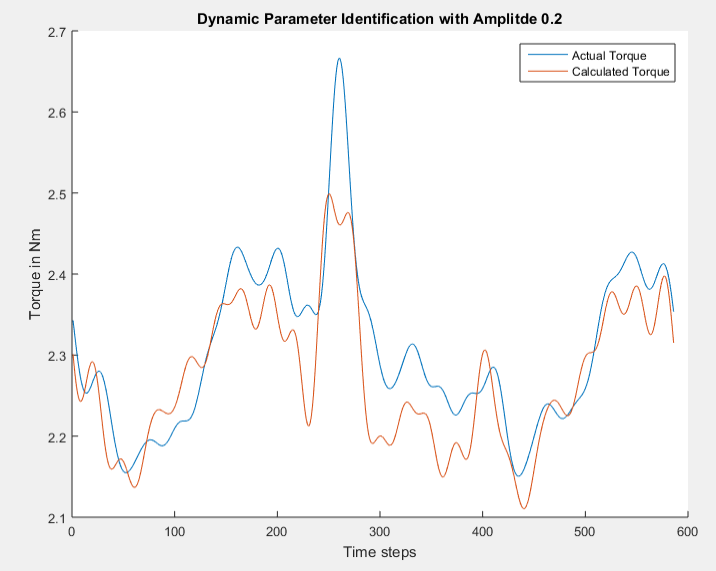

Figure 47 - Calculated and Measured Torque differences for Amplitude 0.2 on Vertical Link

6.3.1.2 Horizontal Amplitude Estimation

Figure 48 shows a dynamic identification run with an amplitude of 0.6 on the horizontal link. The calculated and measured torque on the actuator during the motion pattern line up, but they are small compared to torques produced in the previous chapter 6.3.1.1.

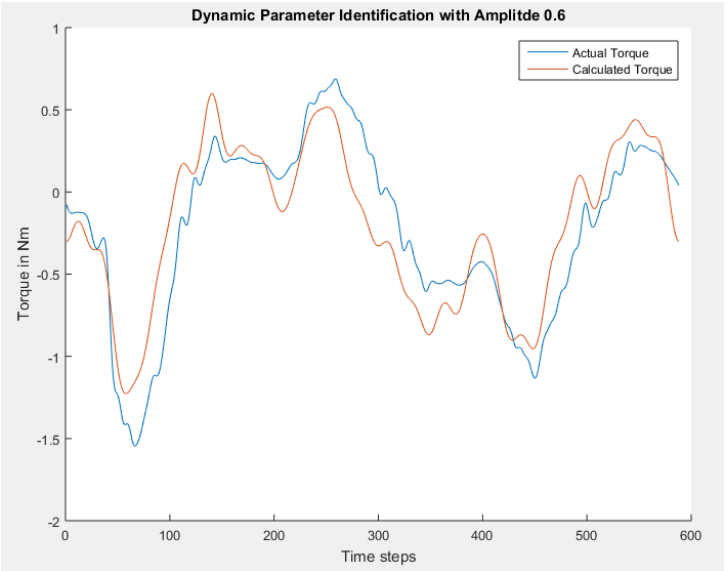

Figure 48 - Calculated and Measured Torque differences for Amplitude 0.6 on Horizontal Link

When using a bigger amplitude of 1.4 in Figure 49, resulting torques are also bigger.

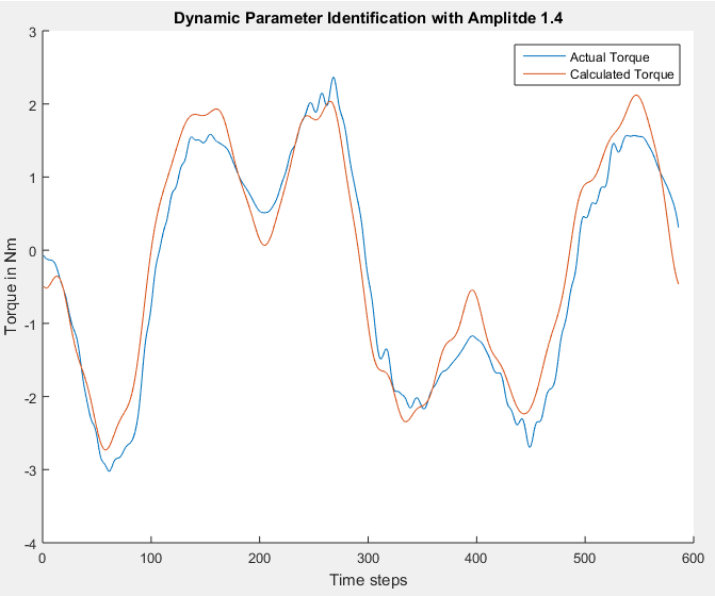

Figure 49 - Calculated and Measured Torque differences for Amplitude 1.4 on Horizontal Link

An amplitude of 2 showed the best outcome (see Figure 50) with bigger resulting torque. In the vertical identification routine, gravity is pulling on the system and putting automatic stress on the torque sensor, resulting in better parameter estimation. In the horizontal case this is not the case, and by having higher amplitudes the torque sensor is better utilized this way.

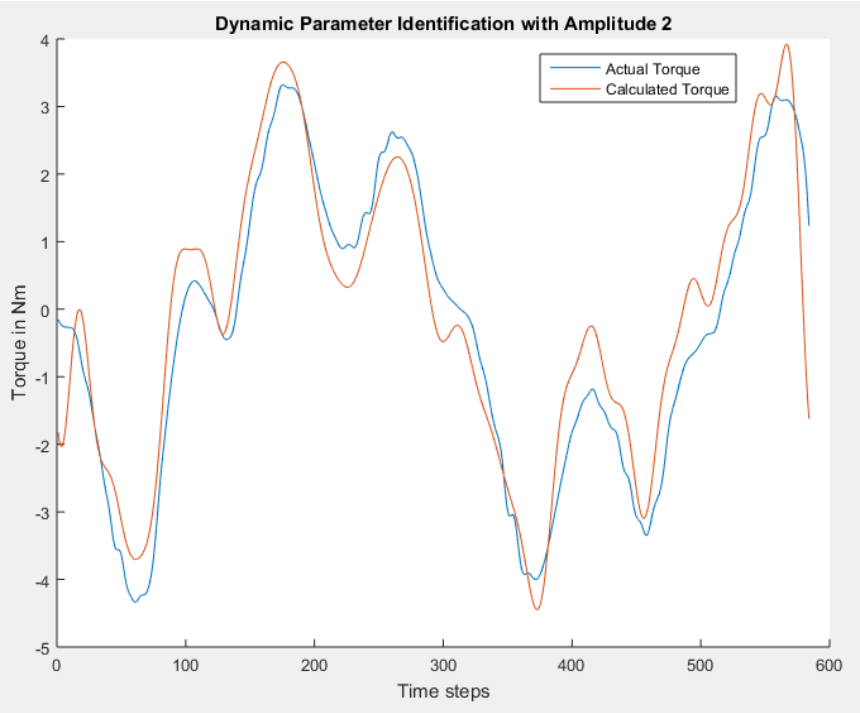

Figure 50 - Calculated and Measured Torque differences for Amplitude 2 on Horizontal Link

6.3.2 Vertical Torque Assistance

Given the outcomes in the previous chapter 6.3.1, the dynamic identification algorithm was performed with an amplitude of 0.2. The estimated inertial parameter vector Φ is listed below:

$$\Phi = \begin{bmatrix} -0.0001 \\ 0.2249 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0725 \end{bmatrix}$$

At first the trajectory is executed on the manipulator supplying position and velocity. The resulting measured torque on the actuator and the calculated torque from the estimated inertial parameters is compared in Figure 51. Real torque values include sensor noise, manipulator jitters and other noise, but the torque strength and also the graph line up.

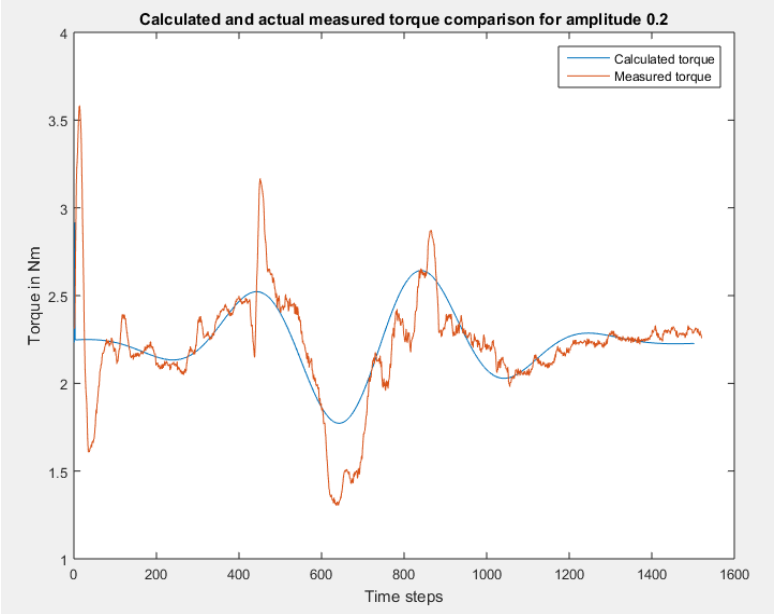

Figure 51 - Calculated and Actual Torque Comparison for Amplitude 0.2 on Vertical Link

To evaluate the calculated torque and torque assistance, this trajectory is now executed on the manipulator with different input commands, PID settings and control strategies. Three different input commands are used:

- Input mode #1: Position
- Input mode #2: Position & Velocity
- Input mode #3: Position, Velocity & Calculated Torque

6.3.2.1 Control Mode Strategy 3 – Position Control

As stated in chapter 2.5.3.1, control strategy 3 is used for position control. Table 7 shows the PID parameters used for three different trials. Only a slight improvement can be seen in

Figure 52 when using additional torque commands on top of position and velocity inputs. It still doesn't quite match the reference trajectory. Both show a better performance than input mode #1 though.

A normal sine trajectory was tested in Figure 53 and Figure 54. PID settings #2 showed no real match for any input mode. The parameters for PID settings #3 were specifically tuned for the sine motion, and input mode #3 allows for a very accurate trajectory compared to the reference.

	Figure 52	Figure 53	Figure 54
PID Settings	#1	#2	#3
Position K_p	4.5	4.5	4.5
Position K_I	0.1	0.1	0.1
Position K_D	1.8	1.8	1.8
Velocity K_P	3.7864	3.7864	3.7864
Velocity K_I	0	0	0
Velocity K_D	1.7	1.7	1.7
Torque K_P	0.12	0.12	0.07
Torque K_I	0.0008	0.0008	0
Torque K_D	0.2	0.2	0.02

Table 7 - PID Settings for CMS3 on Vertical Links

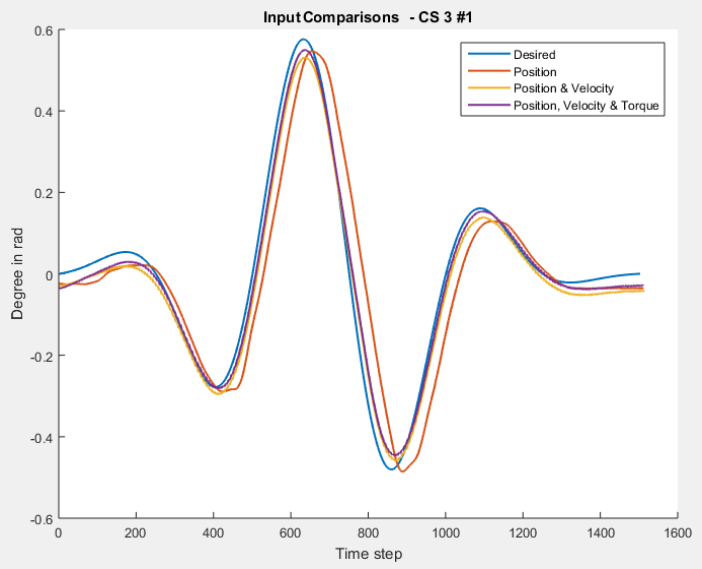

Figure 52 - Input Comparisons on Vertical Link with CMS 3 on PID Settings #1

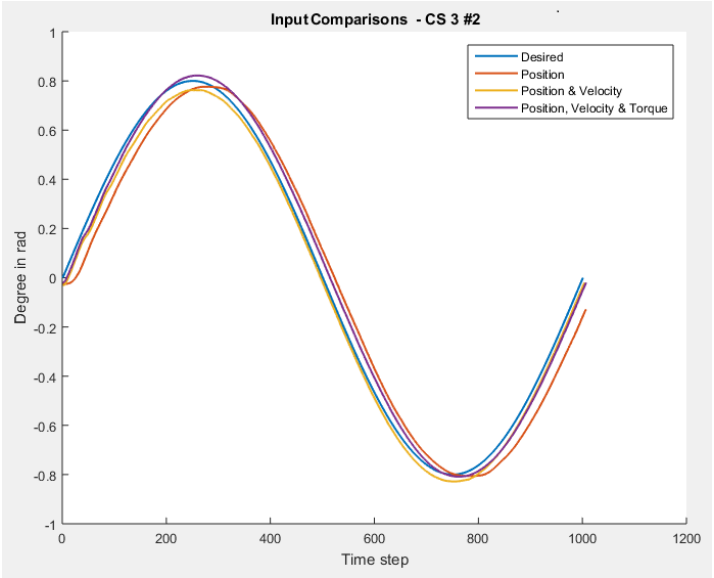

Figure 53 - Input Comparisons on Vertical Link with CMS 3 on PID Settings #2

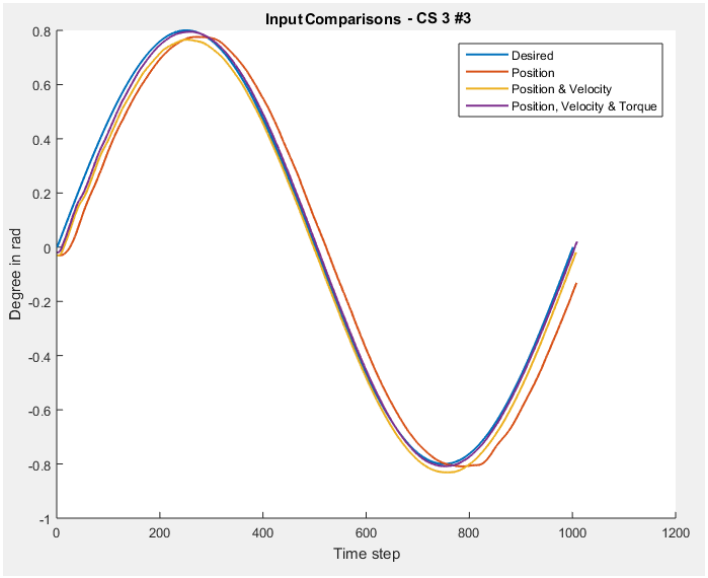

Figure 54 - Input Comparisons on Vertical Link with CMS 3 on PID Settings #3

6.3.2.2 Control Mode Strategy 4 – Torque Control

All three experiments for control strategy 4 use the WSM as motion pattern. The PID settings are listed in Table 8. Position, velocity and Torque input commands produce very good results and follow the reference trajectory very closely in Figure 55, Figure 56, and Figure 57. No PID settings were found to come close to the reference with input mode #1 and #2.

	Figure 55	Figure 56	Figure 57
PID Settings	#4	#5	#6
Position K_p	29	15	21
Position K_I	0	0.001	0
Position K_D	5	4	3
Velocity K_P	0.1864	3.1864	3.1864
Velocity K_I	0	0	1
Velocity K_D	0.1	2.1	2.1
Torque K_P	0.2796	5.12	0.3
Torque K_I	0	0.01	0
Torque K_D	0.9321	2.3321	1.6321

Table 8 - PID Settings for CMS4 on Vertical Links

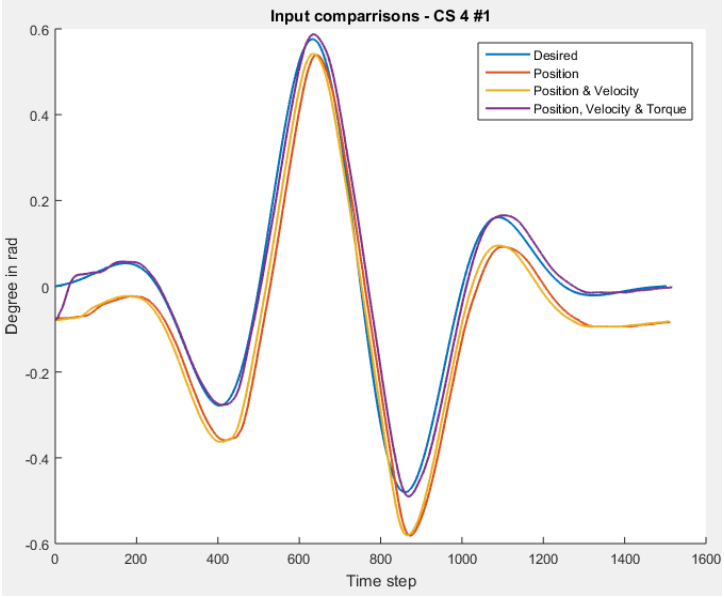

Figure 55 - Input Comparisons on Vertical Link with CMS 4 on PID Settings #4

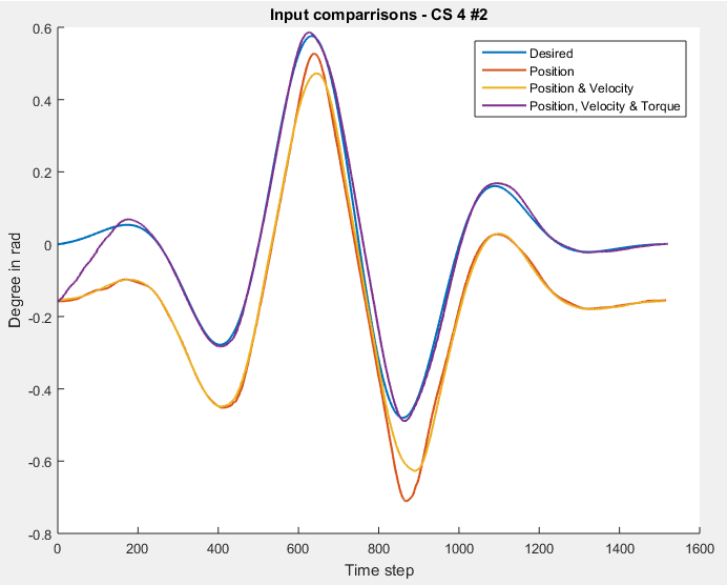

Figure 56 - Input Comparisons on Vertical Link with CMS 4 on PID Settings #5

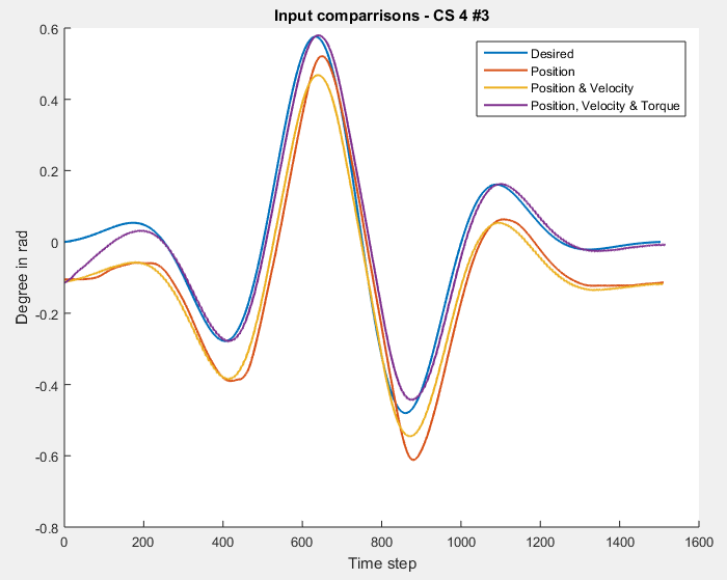

Figure 57 - Input Comparisons on Vertical Link with CMS 4 on PID Settings #6

6.3.3 Horizontal Torque Assistance

Using the results from chapter 6.3.1.2 an amplitude of 2 was used for the dynamic identification algorithm. The resulting inertial parameter vector is as follows:

$$\Phi = \begin{bmatrix} 0.0001 \\ -0.0083 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.1680 \end{bmatrix}$$

Same as in the Vertical Torque Assistance chapter, a WSM is executed to gather calculated and measured torque values as shown in Figure 58. While the torque values for the vertical link started at around 2 Nm, here the values start around the zero mark. This is due to the fact that the actuators have to work against gravity in the vertical position. The calculated torques again line up with the measured torques.

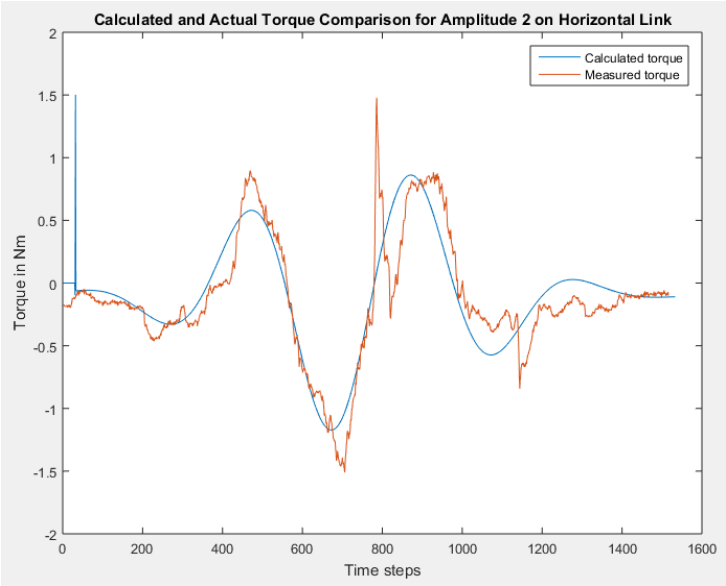

Figure 58 - Calculated and Actual Torque Comparison for Amplitude 2.0 on Horizontal Link

6.3.3.1 Control Mode Strategy 3 – Position Control

For CMS 3 no viable PID parameters could be found to supply velocity values to the actuators. All tested values resulted in instabilities.

6.3.3.2 Control Mode Strategy 4 – Torque Control

The PID parameters for the three trials performing a WSM are listed in Table 9 and are the same as in chapter 6.3.2.2. The differences for the horizontal link are smaller compared to vertical alignment. Input mode #3 again shows good results which follow the reference closely.

	Figure 59	Figure 60	Figure 61
PID Settings	#4	#5	#6
Position K_p	29	15	21
Position K_I	0	0.001	0
Position K_D	5	4	3
Velocity K_P	0.1864	3.1864	3.1864
Velocity K_I	0	0	1
Velocity K_D	0.1	2.1	2.1
Torque K_P	0.2796	5.12	0.3
Torque K_I	0	0.01	0
Torque K_D	0.9321	2.3321	1.6321

Table 9 - PID Settings for CMS4 on Horizontal Links

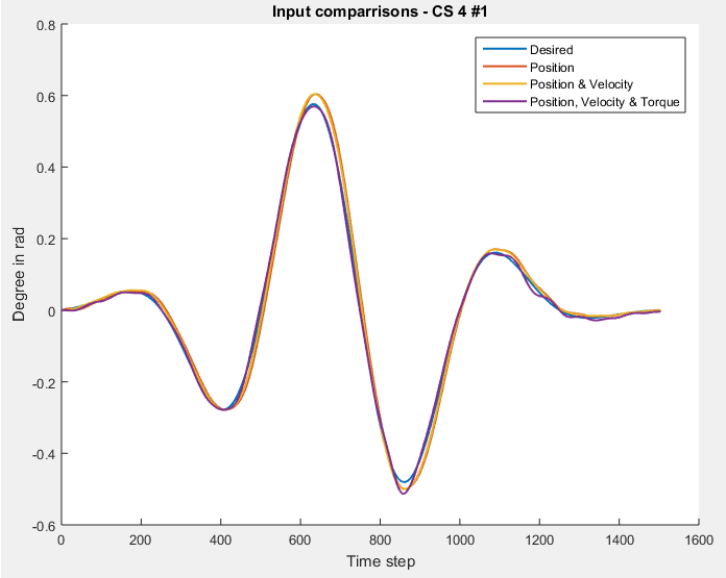

Figure 59 - Input Comparisons on Horizontal Link with CMS 4 on PID Settings #4

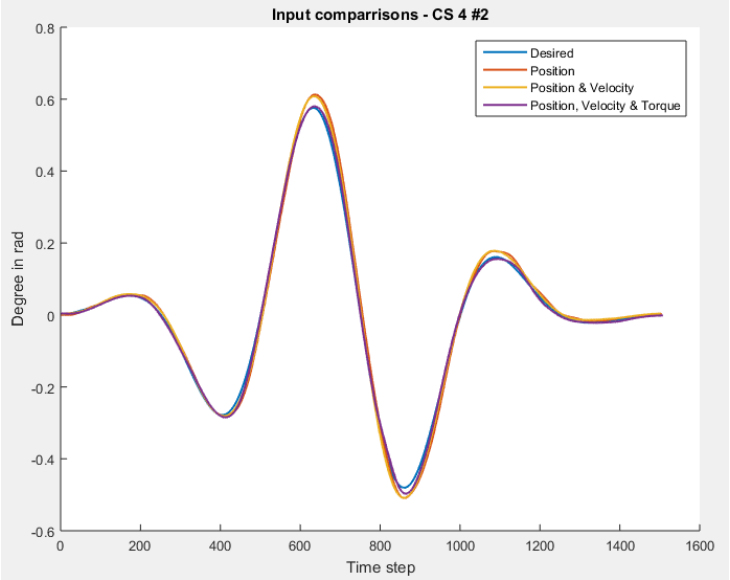

Figure 60 - Input Comparisons on Horizontal Link with CMS 4 on PID Settings #5

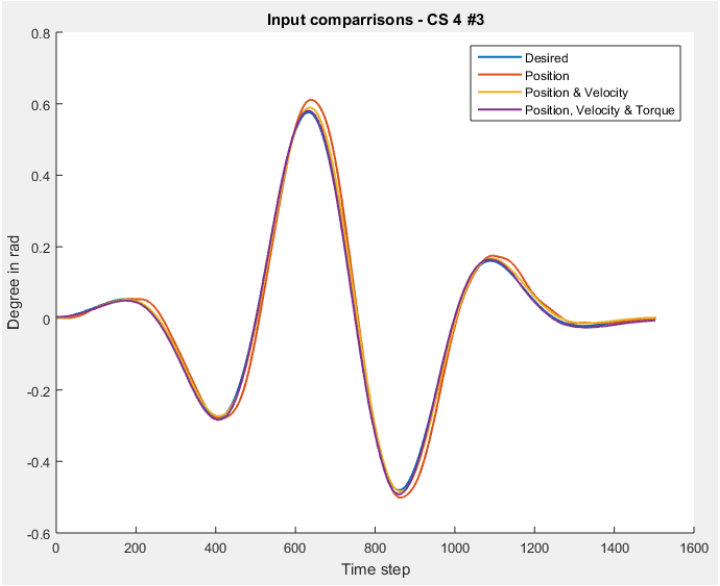

Figure 61 - Input Comparisons on Horizontal Link with CMS 4 on PID Settings #6

6.3.4 Conclusion

In the previous chapters, dynamic identification and torque assistance was successfully performed on horizontal and vertical aligned links. While kinematic parameters in chapter 6.2 can be compared to a ground truth, measuring the inertial parameters of a robot manipulator is difficult. Therefore, the identification effectiveness could only be evaluated by looking at the trajectory accuracy when supplying torque values which are calculated with the help of estimated inertial parameters.

The vertical amplitude estimation showed best results with lower amplitudes. Faster movements actually produced lower torque measurement. Slower movements allow the torque sensors to work better with the forces applied by gravity.

On the other hand, horizontal amplitude estimation resulted in showing better results with higher amplitudes. Since slow movements require low torques and gravity can’t be acting on the sensors, bigger trajectories are needed to fully excite the system and estimate the inertial parameters.

Looking at the torque assistance experiments it can be concluded that finding the right PID parameters for the manipulator setup at hand is very important. Even with suboptimal PID values, adding torque as an input command showed improvements in trajectory movements. Especially CMS 4 with vertical links gain a lot from this approach. Since the inertial parameters are smaller in horizontal position due to the lack of gravity as an extra force on the system, the impact was smaller. PID tuning also showed to be more difficult.

7. Conclusion

This chapter will conclude the work that has been done for this thesis. First, an overall summary of the thesis is presented, and in particular the effectiveness and problems of the parameter identification algorithms is discussed closer. Finally, potential future work is outlined given the results of the experiments.

7.1 Summary

This thesis describes the theoretical background and implementation of kinematic and dynamic parameter identification for a modular robot platform. Furthermore, the algorithms were tested on X5 modules produced by Hebi-Robotics to verify the viability of the procedures.

Chapter 2 introduced concepts related robotics, for example different kinds of robot manipulators. Also the axis-representation was introduced which is in general not as common and used in the kinematic identification algorithm. While kinematics and dynamics are a huge topic, the basics which are required for the thesis were explained. Finally, the modules used in the Biorobotics lab at CMU were described.

Related work was listed in chapter 3 and a brief overview of the papers gives a basis for discussion to why the used procedures were chosen.

Chapter 4 and 5 reason why the given identification schemes are used. For kinematic parameter identification a lot of work has been done in the last decades, but most of it is not applicable for the robotic system used in this thesis. Recent approaches for dynamic parameter identification make use of detailed, complicated models which is not feasible for the current robot setup. Therefore, a fairly old but reliable approach by Atkeson et al. was chosen.

Kinematic and dynamic estimation experiments on X5 modules are performed in chapter 6 with successful results. Link length estimation for horizontal links showed promising results by showing good accuracies and small deviation from the mean. Furthermore, the results agree with theory. On the other hand, vertical estimation only worked properly for link length 484 mm. Dynamic parameter estimation showed good results for both link alignments, though different amplitude parameters have to be used. Torque assistance for vertical links showed more pronounced effects due to gravity putting extra force on the actuators and torque sensors. This resulted in an inertial parameter vector which could produce higher torques.

Reasons for varying results can be attributed to the fragile setup for vertical estimation. During motions joint and link shaking was more visible compared to the sturdier structure for horizontal estimation. Additionally, PID parameters turned out to be very important. Different link lengths or setups can require different PID parameters and suboptimal values can result in jitters, over-swinging, and suboptimal trajectories. The problem is, that tweaking nine different parameters for several different setups is complicated and time consuming. Furthermore, differences in actuator performance made it difficult to find one

setting which worked for all actuators the same. This was especially apparent when no experiments with CMS 3 could be performed on the horizontal torque assistance.

7.2 Future Work

Estimation results were mostly promising for kinematic and dynamic parameter identification. As discussed in the summary chapter 7.1, there are a few problems which should be addressed first in future work.

By using the same new module iteration as mentioned in chapter 6.1 more consistent results could be achieved. After having a homogenous module setup, PID parameters can be tuned more effectively since they will apply for the whole manipulator. Furthermore, a sturdier manipulator setup can definitely improve results. This includes connections between links and joints but also the fixation of the robot base to the ground.

After having the physical problems fixed, and having good general PID parameters, expanding the use of the algorithms is the next step. As stated in chapter 6, the kinematic parameter identification can be used to compute the full kinematic forward model for a several DOF manipulator. If estimation accuracy is dependent on link alignment, the IMUs can be used to automatically detect the gravity vector and thus the location in space. This information can be used to then choose the right parameters for motion patterns.

If length estimations are also very dependent on the correct amplitudes, a set of length amplitude pairs can be found through experiments. After initial length estimation with one motion pattern amplitude for all links, problem links depending on length can be identified and estimated again with a new amplitude. Moreover, several trials of link estimations showed results close together for longer links. To improve performance, links estimation can be run several times to calculate the mean value. Having a manipulator with several links, the forward model can be calculated and then it’s accuracy measured with experiments.

The dynamic identification shows promising results for a 1-DOF arm. A full model must also be tested on a several DOF arm. If results are not satisfying, an approach from chapter 3.2 can be used. To supply a needed model of the system, the estimated kinematic model can be used.

Further improvements can be made on the algorithms used. Instead of moving one joint at a time, performing the CPA for kinematic identification, whole manipulator movements can be utilized. It must be accounted for moving frames and data fusion of the different sensors must be implemented. Furthermore, when performing the CPA, incorporating sensor data from distal joints could show improvements.

7.3 Innovative Aspects

Robotics as a whole is still dominated by big, expensive, and very specialized industrial robotics. Due to new technology and cheaper electronics, a move towards smaller and versatile robotics has been trending.

For now, the field of robotics and handling of robot manipulators still remains a field which requires a lot of specialized knowledge and can only performed by professionals though. Especially performing complicated, error prone math to get the robot to work is currently a challenge. These knowledges in form of kinematic and dynamic models is needed for effectively performing robot manipulation tasks like force control or locomotion. Furthermore, each robot configuration has a unique set of those models. Therefore, to allow for quick reconfiguration, this process needs to be automated as much as possible to match the performance of purpose-built systems.

The kinematic and dynamic parameter identification presented in this thesis is able to extract simple models of the manipulators with good accuracy. Experiment results for horizontal links showed length estimation errors below 5 %. Vertical link length estimation showed a worse performance, but this can be attributed to a suboptimal manipulator setup and inconsistency in the used modules. After improvements in those areas, estimation accuracy should be comparable with the horizontal counterpart. The dynamic identification of the inertial parameters of robot manipulators was used to calculate supplementary torques which are then sent to motors as additional input command. These torque values allow for compensation of forces on the joints and links due to inertia and gravity. Experiments showed improvements in trajectory accuracy in all setups when supplying the calculated torque.

The work in this thesis set a basis for easier reconfiguration of modular robotic manipulators. Experiment results showed, that both the kinematic and dynamic identification schemes show good results and are feasible to extract models for control of the robot. Instead of manually calculating manipulator models, the robot can run identification patterns on its own to calculate the important parameters.

This opens new areas of uses for robots, and new opportunities for research. This includes areas where modular robotics can lead to improvements efficiency and costs. Depending on changing requirements, independent hardware components can be quickly assembled for individual tasks, just like combining Lego bricks. Agile manufacturing can be achieved by quickly adapting and reconfiguring the robot to solve a different range of tasks, depending on the task at hand. Furthermore, mobile robots for search-and-rescue or e.g. future robots for Mars missions can benefit from automatic parameter identification. In case of hardware wear or deformation, the robot can reassess its parameters for optimal control. This can lead to self-reconfiguration, self-repairing, self-assembling, error detection and error handling.

8. Bibliography

[1] T. R. Kurfess, Robotics and Automation Handbook, Taylor & Franci, 2005.

[2] M. W. Spong and et al., Robot Dnamics and Control - Second Edition, 2004.

[3] "kuka-robotics," [Online]. Available: http://www.kuka-robotics.com/en/products/industrial_robots/low/kr16_2/start.htm. [Accessed 21 August 2016].

[4] B. Siciliano et al., Robotics - Modelling, Planning and Control, Springer, 2010.

[5] S. Belongie, "wolframalpha," [Online]. Available: http://mathworld.wolfram.com/RodriguesRotationFormula.html. [Accessed 24 August 2015].

[6] G. Besold, "Memphys," [Online]. Available: http://www.memphys.sdu.dk/~besold/INDEX/axis-angle.pdf. [Accessed 24 August 2015].

[7] R. N. Jazar, Theory of Applied Robotics - Kinematics, Dynamics, and Control, Second Edition, Springer, 2010.

[8] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," in *Journal of Applied Mechanics*, 1955, pp. 215-221.

[9] D. Morris, "techhouse," [Online]. Available: https://www.techhouse.org/~dmorris/projects/tutorials/inertia.tensor.summary.pdf. [Accessed 23 August 2016].

[10] "Biorobotics Laboratory," [Online]. Available: http://biorobotics.ri.cmu.edu/robots/index.php. [Accessed 14 July 2016].

[11] "Biorobotics Lab," [Online]. Available: http://biorobotics.ri.cmu.edu/applications/nuclearApplication.php. [Accessed 14 July 2016].

[12] G. A. Pratt and M. M. Williamson, "Series Elastic Actuators," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995, p. 399–406.

[13] D. Rollinson et al., "Design and Modeling of a Series Elastic Element for Snake Robots," in *ASME Dynamic Systems and Control Conference (DSCC)*, 2013.

[14] D. Rollinson et al., "Design and Architecture of a Series Elastic Snake Robot," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4630 - 4636, 18 September 2014.

[15] H. Robotics, "X-Series Industrial Smart Actuator Datasheet," http://hebirobotics.com/, 2016.

[16] hebirobotics, "hebirobotics," [Online]. Available: https://hebirobotics.atlassian.net/wiki/display/documentation/Control+Modes. [Accessed 28 August 2016].

[17] B. Siciliano and O. Khatib, "Model Identification," in *Handbook of Robotics*, Springer, pp. 321-344.

[18] B. Mooring, M. Driels and Z. Roth, Fundamentals of Manipulator Calibration, John Wiley and Sons, 1991.

[19] C. W. Wampler and J. M. Hollerbach, "The Calibration Index and Taxonomy for Robot Kinematic Calibration Methods," *The International Journal of Robotics Research*, p. 573–591, 1996.

[20] M. Hoffmann et al., "Body schema in robotics: A review," *IEEE Transactions on Autonomous Mental Development*, p. 304–324, 2010.

[21] J. Sturm, C. Plagemann and W. Burgard, "Adaptive body scheme models for robust robotic manipulation," in *Robotics: Science and Systems IV*, MIT Press, 2008.

[22] e. a. Ulbrich, "Rapid Learning of Humanoid Body Schemas with Kinematic Bezier Maps," in *9th IEEE-RAS International Conference on Humanoid Robots*, 2009.

[23] I. Ha, "Kinematic parameter calibration method for industrial robot manipulator using the relative position," in *Journal of Mechanical Science and Technology 22*, 2008, pp. 1084-1090.

[24] G. Du, P. Zhang and D. Li, "Online robot calibration based on hybrid sensors using Kalman Filters," in *Robotics and Computer-Integrated Manufacturing 31*, 2015, pp. 91-100.

[25] G. Du and P. Zhang, "Online Serial Manipulator Calibration Based on Multisensory Process Via Extended Kalman and Particle Filters," in *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 2014.

[26] G. Canepa, J. M. Hollerbach and A. J. M. A. Boelen, "Kinematic Calibration by Means of a Triaxial Accelerometer".

[27] P. Zhang and G. Du, "IMU-Based Online Kinematic Calibration of Robot Manipulator," in *The Scientific World Journal*, Hindawi Publishing Corporation, 2013.

[28] Y. Lin, H. Min and H. Wei, "Inertial measurement unit–based iterative pose compensation algorithm for low-cost modular manipulator," in *Advances in Mechanical Engineering*, 2016.

[29] P. Mittendorfer, E. Dean and G. Cheng, "Automatic Robot Kinematic Modeling with a Modular Artificial Skin," in *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 749 - 754.

[30] H. Hage, P. Bidaud and N. Jardin, "Practical consideration on the identification of the kinematic parameters of the Staubli ¨ TX90 robot," in *13th World Congress in Mechanism and Machine Science*, 2011.

[31] D. J. Bennet and J. M. Hollerbach, "Autonomous Robot Calibration for Hand-Eye Coordination," in *The International Journal of Robotics Research* , 1991.

[32] A. Calanca and et al., "MIMO Closed Loop Identification of an Industrial Robot," in *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 2011.

[33] C. H. An, C. G. Atkeson and J. M. Hollerbach, "Estimation of Inertial Parameters of Rigid Body Links of Manipulators," in *Proceedings on Decision of 24th and Conference on Decision Control*, 1985.

[34] I. D. Landau, B. D. O. Anderson and F. De Bruyne, "Closed-loop output error identification algorithms for nonlinear plants," in *IEEE Conference on Decision and Control*, 1999, pp. 606-611.

[35] M. Gautier, A. Janot and P.-O. Vandanjon, "A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics," in *IEEE Transactions on Control Systems Technology Vol. 21*, 2012, p. 428-444.

[36] M. Gautier and S. Briot, "Dynamic Parameter Identification of a 6 DOF Industrial Robot using Power Model," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[37] M. E. Skalar, "Geometric calibration of industrial manipulators by circle point analysis," in *Proc 2nd Conf Recent Adv Robot*, 1989, p. 178–202.

[38] C. G. Atkeson, C. H. An and J. M. Hollerbach, "Rigid Body Load Identification for Manipulators," in *Proceedings on Decision and of 24th Conference on Decision Control*, 1995.

[39] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd Edition, Pearson/Prentice, 2005.

[40] P. Mittendorfer, E. Dean and G. Cheng, "Automatic Robot Kinematic Modeling with a Modular Artificial Skin," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 749-754.