Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Aslak J. Strand

Control and Sensor Systems for Offshore and Aquaculture Cranes

Master's thesis in TPK4960 - Robotics and Automation
Supervisor: Christian Holden
Co-supervisor: Sveinung Ohrem
June 2021

NTNU
Norwegian University of
Science and Technology

Aslak J. Strand

Control and Sensor Systems for Offshore and Aquaculture Cranes

Master’s thesis in TPK4960 - Robotics and Automation
Supervisor: Christian Holden
Co-supervisor: Sveinung Ohrem
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Institutt for Maskinteknikk og Produksjon
Fakultet for ingeniørvitenskap
NTNU – Norges teknisk-naturvitenskapelige universitet

TILGJENGELIGHET
Ugradert

MASTEROPPGAVE 2021

FAGOMRÅDE: Robotteknikk	DATO: 10.06.2021	ANTALL SIDER: 33

TITTEL:

Regulering og Sensor Systemer for Kraner i Offshore og Akvakultur

Control and Sensor Systems for Offshore and Aquaculture Cranes

UTFØRT AV:

Aslak J. Strand

SAMMENDRAG:

Ettersom offshore industrien er like viktig i dag som tidligere, er det nødvendig med ny og forbedret teknologi for å håndtere mulige problemer som kan oppstå. Derfor vil vi igjennom denne oppgaven se på muligheten for å lage et stabiliserings-system for last som henger fra en kran.

Offshore kranoperasjoner kan være utsatt for kraftig vind og bølger, noe som kan forårsake store svingninger på en hengende last. Ved å redefinere kransystemet til å fungere som en robotarm, for så å legge inn ett system for å motvirke svingningene, vil man kunne gjennomføre tryggere løfteoperasjoner til sjøs.

I løpet av denne oppgaven vil vi se på oppbygning og regulering av robotarmer, samt undersøke hvorvidt dataen fra billige IMUer er stabil nokk til å brukes for reguleringen.

FAGLÆRER: Christian Holden

VEILEDER(E): Sveinung Ohrem

UTFØRT VED: NTNU

Department of Mechanical and Industrial Engineering
Faculty of Engineering
NTNU – Norwegian University of Science and Technology

ACCESSIBILITY
Unclassified

MASTER THESIS 2021

SUBJECT AREA: Robotics	DATE: 10.06.2021	NO. OF PAGES: 33

TITLE:

Control and Sensor Systems for Offshore and Aquaculture Cranes

Regulering og Sensor Systemer for Kraner i Offshore og Akvakultur

BY:

Aslak J. Strand

SUMMARY:

As the aquaculture industries are as important as ever, there's a constant need for new and improved technology to counter possible issues. We will therefore investigate the possibilities of load stabilizing control systems for offshore and aquaculture cranes in this thesis.

Offshore crane operations may often be subject to high winds and waves, whereas the suspended load will start oscillating. By redefining the crane system as a robotic manipulator, and then implementing a control structure to counteract these dynamic disturbances, one may be able to carry out safer and more controlled operations.

Throughout this thesis we are looking into how to build and control robotic manipulators, and whether data from affordable IMUs are reliant enough to be used as an input to the control structure.

RESPONSIBLE TEACHER: Christian Holden

SUPERVISOR(S): Sveinung Ohrem

CARRIED OUT AT: NTNU

Kunnskap for en bedre verden

Department of Mechanical and Industrial Engineering

TPK4960 - Robotics and Automation, Master's Thesis

Control and Sensor Systems for Offshore and Aquaculture Cranes

Author:
Aslak J. Strand

10.06.2021

[Page intentionally blank]

Preface

As the aquaculture industries are as important as ever, there is a constant need for new and improved technology to overcome potential issues. One of these issues is the dynamic and constantly unstable environment. Without any landscape to dampen the effect of wind and waves, the boats, crew and equipment are highly affected by the almost constant motion they endure. This leads some offshore operations to be more difficult and dangerous than necessary.

With a background in satellite engineering, I can understand the need and importance of stabilizing systems in the offshore business. Whereas there are several different types of stabilization systems already in use, such as heave compensation, dynamic positioning systems, and anti-roll tanks, most of these systems are both expensive and difficult to implement. This is what led my interest in stabilizing systems for on-board cranes, whereas these could potentially make offshore crane operations safer, and they could be made and implemented at a potentially lower cost.

Throughout the project I have learned a lot about offshore operations, as well as a lot more within the fields of robotics, especially and how to build and control robots. These skills will be valuable going forward, as robotics and control structures are constantly being implemented in everyday systems.

Some of the measurements in this paper could not have been conducted without the help of Sintef and Dr. Ohrem, as well as my supervisor Dr. Holden at NTNU, and I would therefore take the opportunity to extend my gratitude for any help, both theoretical as well as in the form of physical access to testing locations.

Aslak J. Strand

Contents

List of Figures

List of Tables

Nomenclature

ARC	Adaptive Robust Controller
DH	Denavit-Hartenberg
EE	End Effector
IMU	Inertial Measurement Unit
I2C	Inter-Integrated Circuit, a multi-user serial communication bus.
IMU	Inertial Measurement Unit
Pitch	Rotation around the y-axis
Roll	Rotation around the x-axis
URDF	Universal Robotic Description Format
Yaw	Rotation around the z-axis

1 Introduction

In this paper we will look into the possibilities of a control structure to stabilize a suspended load from a crane, using IMU measurements to observe how the load is moving. The study is an extension of a previously completed work, whereas the goal was to design a functional computer simulated Boom and Knuckle crane. This crane model will become a key component to arriving to a control structure that could help stabilize load suspended from a crane on a movable platform.

The basis for the control structure is a implementation of robotics theory, whereas the crane itself will be operated as a semi-automatic robot manipulator. This renders the crane able to receive and act on commands from an operator or sensors, thus yielding it a a good basis for implementing automatic control systems.

As most cranes today are controlled by individually actuating the different joints, the crane operator has to account for load oscillations by manually configure each joint such that the crane counteracts these motions. Aquaculture cranes tend to be highly impacted by the environment, whereas waves and wind becomes large disturbances for the suspended load. By implementing a system to automatically control the oscillations, the crane operations may become easier to control in such environments.

There is also a desire to evaluate the use of IMUs, and to see the impact of switching out expensive and precise IMUs with cheaper ones that are more prone to bias and drift. If one were able to decrease the cost of measurement units, a control structure would be a lot easier to implement in a real life scenario.

By methodically adding new features to the existing simulation step by step, the control system will be gradually improved throughout the project. Rendering a final control system that may stabilize the load based on realistic measurement data.

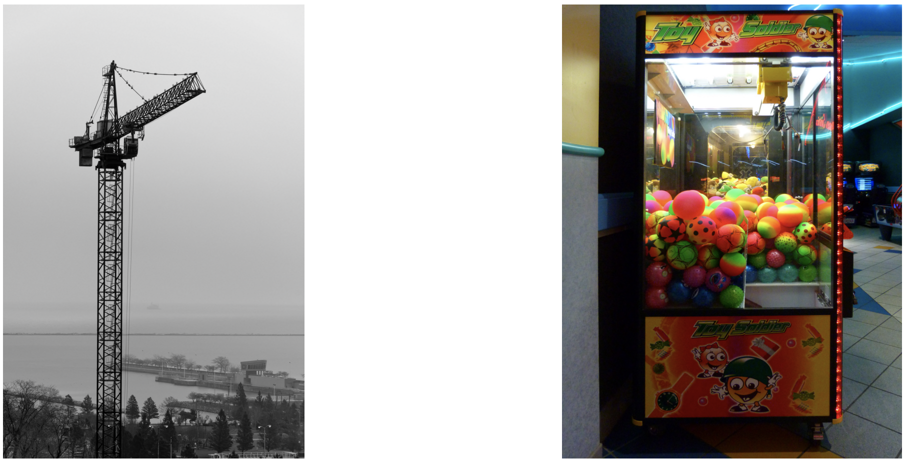
2 Specialization Project

In the following section a series of important notes from the specialization project (*1*) will be presented such that the reader will possess an adequate level of understanding of what has been done prior to this report, what the intended end goal is, and how to achieve this goal. The notes themselves should not be seen as a part of the master theses, yet it might be useful to understand the previously completed work. For the full report, please see appendix 1.

2.1 Literature Study

When imagining a crane system, a large, fixed and sturdy construction used for displacing equipment and materials comes often to mind, whereas wind is often the unknown factor for how the cargo will move. However, cranes comes in various shapes and sizes, as well as they are rigged on different foundations such as concrete, wheels or vessels. Crane systems are not only limited to land based operations, but highly applicable in the offshore business. Offshore cranes may look like a land based crane, and be operated in the same manner, yet they are subject to a few more disturbances that will impact the trajectory of the cargo.

The definition of a crane, according to the Big Norwegian Encyclopaedia, is a machine for lifting and displacing loads, both vertically and horizontally. This definition seem to be quite broad, however, as cranes comes in a variety of shapes and sizes, specified only by the job they are crated for solving. May it be tall “classical” tower cranes for building skyscrapers as seen in Figure 1a, or smaller machinery such as the claw in a Claw Machine, as seen in Figure 1b, they will all fall under the same definition. As seen in the figures, the differences in size and shape are fairly large, yet they operate in similar manner whereas a hoisting mechanism is able to be moved up and down, while the suspension may move in a fixed set of axis or rotational degrees.



(a) Tower Crane.

(b) Claw Machine.

Figure 1: Various crane designs.

Most land based cranes are stationary with little to no other disturbances other than wind. However when it comes to offshore lifting operations, there are more disturbances to take into account when a crane is in use. As with stationary cranes you will encounter winds when working offshore. The difference is that over landmasses there is more friction to slow down the wind speeds, which is not the case offshore. As there are limited friction, the wind speeds during offshore hoisting operations may be far greater than that of land operations, which may induce a more irruptive disturbance.

Furthermore offshore hoisting operations are usually completed from a non-stationary platforms such as a vessel. These platforms are subject to the dynamics of the ocean, whereas waves and streams will have an effect on the end effector of the crane, both in heave elevation and in a roll, pitch and yaw movement. These different movements can be seen in Figure 2.The extra disturbances to the crane yields a more difficult operation compared to that of a fixed land based crane. This is leading to a higher level of skill required from the operators in order to safely complete the task at hand, either it is submerging or raising of subsea installations, resupply between vessels, or maintenance or other operations on structures or instalments at sea.

The main difficulties with offshore craning usually present them selves when using the offshore cranes and the deck cranes, i.e. not when the vessel is docked, and these difficulties are twofold. Firstly the dynamic environment will affect the hanging load in unpredictable ways, making it harder to move and place the load safely and correctly, the movement in the load induced from the environment may also be large enough to further induce extra movement of the vessel, drastically increasing the disturbance. The second problem appears as a mechanical result of the extra dynamic factors, whereas the shear strength of the crane itself and the strength of lifting gears and motors are being pushed towards their limits due to extra forces on the load.

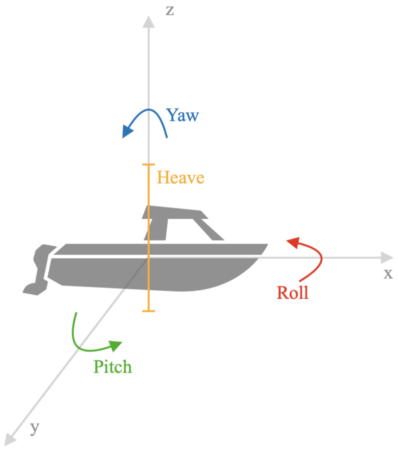
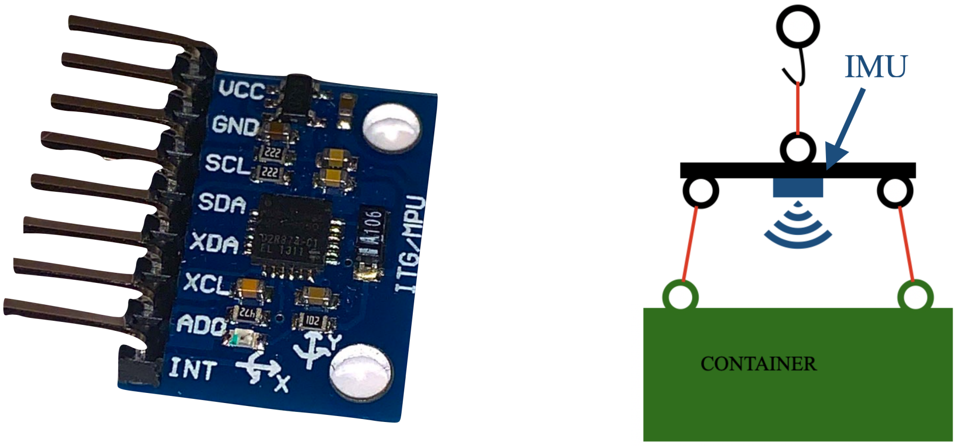


Figure 2: Roll, pitch, yaw and heave.

In order to successfully control the position of the load hanging from a wire, two possible methods comes to mind. One way would be using Adaptive Robust Controllers (ARC) in order to counteract the disturbances. These ARCs are designed to adapt to uncertainty, yielding them reasonably good for significant and random disturbances. They are however a more expensive and harder to implement, thus reducing the cost effectivity if you were to implement such systems on a larger scale.

Another way of controlling the hanging load could be by using an Inertial Measurement Unit (IMU) directly or close by the load. These IMUs are small and relatively cheap pieces of hardware that outputs data related to movement. A basic IMU, such as the MPU6050 chip in Figure 3a, outputs data such as acceleration in the translation axis and angular velocity around the axis. This will yield the possibility to track the speed of the load. By placing this IMU on the lifting beam as seen in Figure 3b, it will stay fairly close to the hanging load, and thus output of the approximate movement of the load it self. These data could then be used to calculate the relative movement between the end effector of the crane and the load in a fast and cheap way. By using this data as an input to the crane controllers, you can effectively set the joystick reference to control the movement of the load it self, thus letting the crane work against disturbances on the load in the translation axis and rotations around the crane suspension.

This would serve as an relatively easy and affordable way of controlling the hanging load, and given that the IMUs are fairly priced they could even be connected to a wireless transfer unit, i.e. a bluetooth or wifi card, and then added directly to any load that is to be moved in order to receive even more accurate movement data of the load. By further investigating this method, the end goal is to find an simple and elegant way of implementing IMU data into the control structure of the crane, such that the joystick input would yield a reference to the movement of the load and not the load suspension at the end of the crane.



(a) MPU6050 IMU chip.

(b) IMU on lifting beam.

Figure 3: Tracking Source

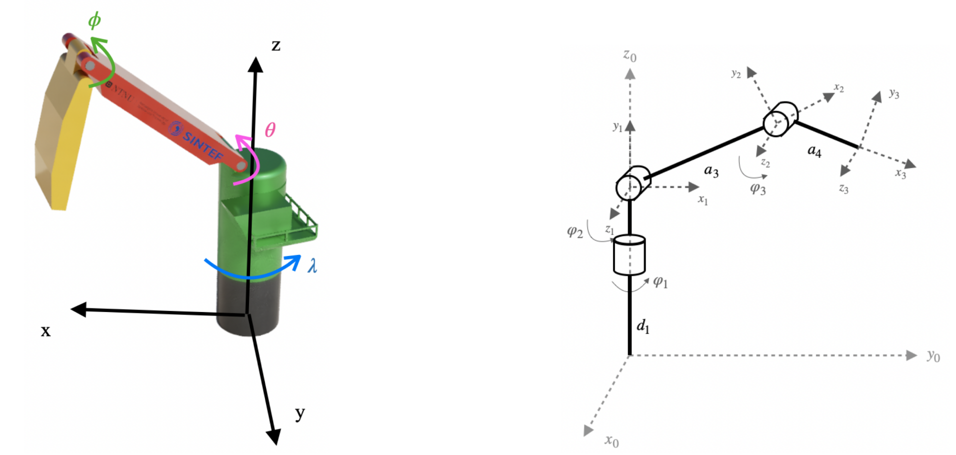
2.2 Modeling

The modelling of the crane will be undertaken as a robotic arm manipulator, in accordance with the Denavit–Hartenberg (DH) parameters. In order to obtain a basic understanding for robotic arm manipulation, a basic crane model will be modelled first, such as shown in Figure 4a, without wire and a hanging load. This manipulator will have the following links; base (black), tower (green), main boom (red) and outer boom (yellow). There is also three revolving joints in the model; between base and tower (λ), between tower and main boom (θ) and between main and outer boom (ϕ).

This design can then be split into DH-parameters, as seen in Table 1. Here each link is represented with the following parameters:

- a_i : Shortest distance between O_i and O_{i+1}, measured along the x-axis.
- α_i : Angle offset for the z-axis of link l_{i-1} and link l_i.
- d_i : Length between link l_{i-1} and link l_i along the z-axis of l_{i-1}. (Prismatic variable)
- θ_i : Angle about l_{i-1} z-axis, from old x-axis to new x-axis. (Revolute variable)
- Joint: Type of joint between link l_i and link l_{i-1}.

A simple illustration of the DH-parameters can be seen in Figure 4b, whereas the cylindrical shapes illustrates revolving joints.



(a) Boom and Knuckle model.

(b) Illustration of DH-parameters.

Figure 4: Model Illustrations.

Table 1: DH-parameters of model.

Link	a_i	α_i	d_i	θ_i	Joint
l_1	0	0	d_1	θ_1	Revolute
l_2	0	$\frac{\pi}{2}$	0	θ_2	Revolute
l_3	a_3	0	0	θ_3	Revolute
l_4	a_4	0	0	0	Fixed

These DH-parameters can then be used to obtain the rotation-matrix T, which describes the offset and orientation of the end effector reference frame relative to the reference frame of the base. The formula for the T-matrix is shown in equation 1, whereas the individual -matrices for each joint can be calculated according to equation 2.

$$T_4^0(q) = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \tag{1}$$

$$A_i^{i-1} = \begin{bmatrix} cos(\varphi_i) & -sin(\varphi_i)cos(\alpha_i) & sin(\varphi_i)sin(\alpha_i) & a_icos(\varphi_i) \\ sin(\varphi_i) & cos(\varphi_i)cos(\alpha_i) & -cos(\varphi_i)sin(\alpha_i) & a_isin(\varphi_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

As the end goal is to control the hanging load suspended in a wire from the end effector of the crane, a more complex model has to be be made. One can assume that an adequate heave compensation has been implemented, thus resulting in a tense wire without slack. We would also need to factor in the prismatic joint that mimics the hoisting mechanism of the wire, in order for the simulated crane to lift or drop the load as it sees fit in order to meet the reference set point.

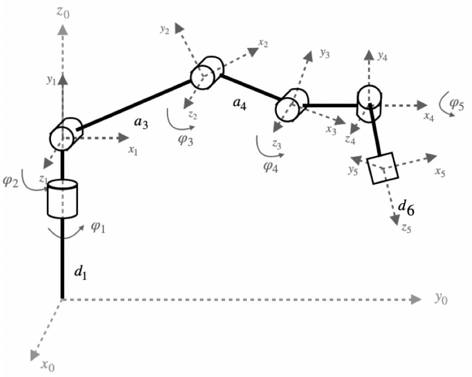


Figure 5: Illustration of complex model.

By appending the new links and joints to the DH parameter table, the resultant DH parameter table is as shown in Table 2, and the updated illustration of the model is shown in Figure 5, whereas a square now represents the prismatic hosting joint, with it's extension axis along the prismatic z-axis.

Table 2: DH-parameters of model.

Link	a_i	α_i	d_i	θ_i	Joint
l_1	0	0	d_1	θ_1	Revolute
l_2	0	$\frac{\pi}{2}$	0	θ_2	Revolute
l_3	a_3	0	0	θ_3	Revolute
l_4	a_4	0	0	θ_4	Revolute
l_5	0	$\frac{\pi}{2}$	0	θ_5	Revolute
l_6	0	$-\frac{\pi}{2}$	d_6	0	Prismatic

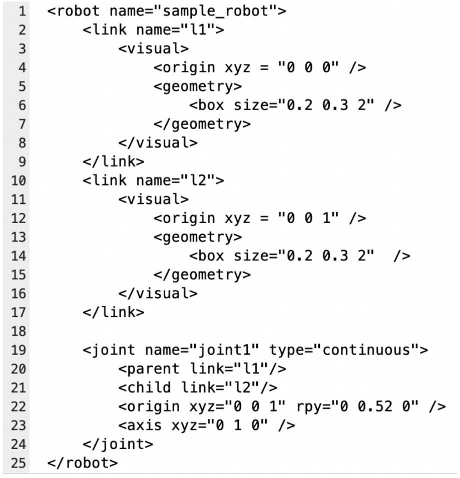
Furthermore, the rotation matrix will then be calculated as seen in equation 3.

$$T_6^0(q) \ = \ A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5 \ = \ T_4^0(q) \cdot A_5^4 \cdot A_6^5 \tag{3}$$

$$T_6^0(q) = \begin{bmatrix} cos\left(\varphi_1 + \varphi_2\right) cos\left(\varphi_3 + \varphi_5\right) & -cos(\varphi_1+\varphi_2)sin(\varphi_3+\varphi_5) & sin(\varphi_1+\varphi_2) & cos(\varphi_1+\varphi_2)cos(\varphi_3)(a_3+a_4) \\ sin\left(\varphi_1 + \varphi_2\right) cos\left(\varphi_3 ? \varphi_5\right) & -sin(\varphi_1+\varphi_2)sin(\varphi_3+\varphi_5) & -cos(\varphi_1+\varphi_2) & sin(\varphi_1+\varphi_2)cos(a_3)(a_3+a_4) \\ sin\left(\varphi_3 + \varphi_5\right) & cos(\varphi_3+\varphi_5) & 0 & d_1+a_3sin(\varphi_3)+a_4sin(\varphi_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 Simulation

There are various ways of implementing a robotic manipulator into Simulink, whereas the most intuitive way is by utilising the Simscape^{TM} toolbox, whereas you may add or remove joints or links by adding the correct blocks to the Simulink diagram. Furthermore, there is the possibility of modelling the robot through the use of a Universal Robotic Description Format (URDF) file, where you can through descriptive code tell Simscape how the basic layout of the robot should be. A basic URDF sample code for a two-link arm can be seen in Figure 6.

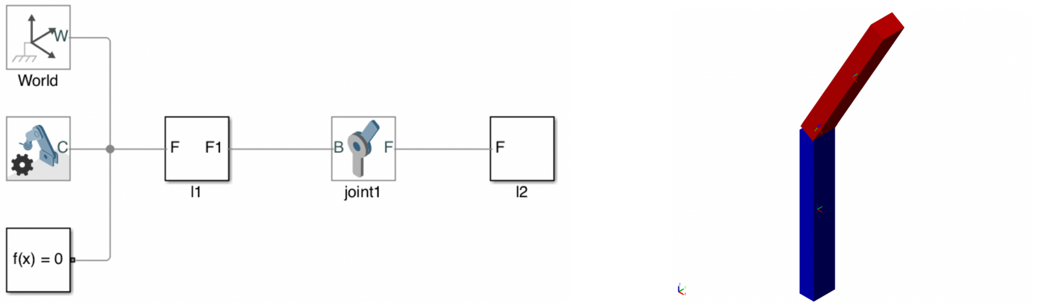


```xml
<robot name="sample_robot">
    <link name="l1">
        <visual>
            <origin xyz = "0 0 0" />
            <geometry>
                <box size="0.2 0.3 2" />
            </geometry>
        </visual>
    </link>
    <link name="l2">
        <visual>
            <origin xyz = "0 0 1" />
            <geometry>
                <box size="0.2 0.3 2"  />
            </geometry>
        </visual>
    </link>

    <joint name="joint1" type="continuous">
        <parent link="l1"/>
        <child link="l2"/>
        <origin xyz="0 0 1" rpy="0 0.52 0" />
        <axis xyz="0 1 0" />
    </joint>
</robot>
```

Figure 6: URDF sample code.

By fully describing the robot this way, you can then implement this robot into Simulink and Simscape by using the MatLab function "smimport(urdf-file)". This command takes the basic robotic arm layout from the URDF file, and generates a set of Simscape blocks to build an adequate representation the code within Simulink, as seen in Figure 7a which then is displayed within the simulation as seen in Figure 7b.



(a) Simulink Model.

(b) Simulated Representation.

Figure 7: Two-link sample model.

One of the main advantages of using the Simscape environment is that each joint-block has a built in possibility of adding torque to the manipulator, as well as several different measurements such as rotation speed, position, angle and acceleration. By utilising these solutions, a state feedback can be made for each joint, thus resulting a system where a feedback controller can be implemented for each subtask of the crane.

By using the Denavit–Hartenberg parameters from Table 2 to describe the robotic manipulator for the crane in URDF, you can describe the length of the links, standard positions and angles, and which type of joint to use between the links. By then converting the URDF file to Simscape, and opting to manually input actuation for the joints as well as opting for position and velocity measurements, a functioning simulation model of the crane will been made. This model can be seen in Figure 8. Note that the final two revolute joints that simulates the suspension point, and the prismatic joint for the wire has been exchanged for a spherical telescoping-joint, which is able to rotate in any direction as well as hoisting the load along its z-axis. The hanging load is modelled as a 10-foot shipping container in order to keep the movement as realistic as possible.

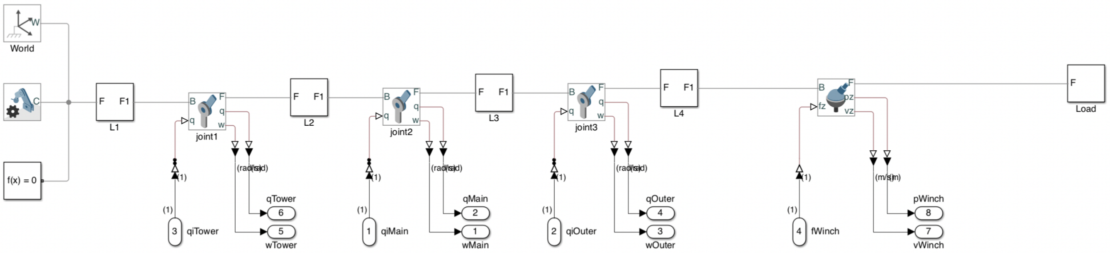


Figure 8: Crane simulink model.

As the crane it self is implemented, the model requires a control structure in order to actuate the joints. This can be achieved using an individual Simulink PID-regulator block for each joint, and using the sensor data from the joint as feedback to the controller. The controller takes into account a setpoint value and the sensor data, and then modifies the input signal to the joint in order to achieve equality between the setpoint and the sensor data. The setpoint value is provided by a user interface, in which two joysticks provides the setpoint for the four different controllers depending on which axis, (x) or (y), the operator moves. Here the left-hand joystick provides the setpoint for the main (x) and outer (y) boom, while the right-hand joystick provides the setpoint for the tower rotation (x) and the winch (y). The full Simulink implementation can be seen in Figure 9.

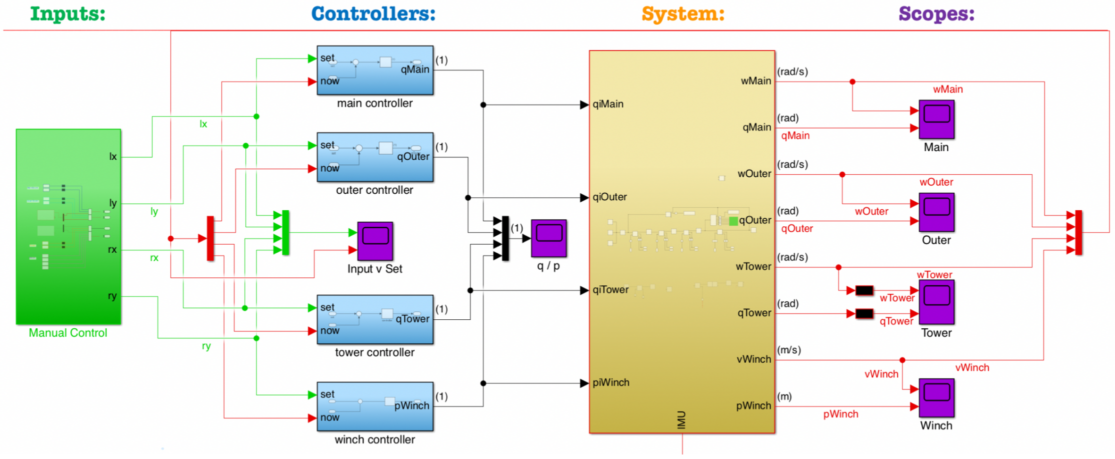


Figure 9: Full simulink implementation.

2.4 IMU Implementation

One of the challenges for this task was to see if the hanging load could be controlled by using acceleration and gyroscope data from an IMU located near or on the load. This IMU would then approximately yield the relative movement of the hanging load in respect to the suspension point on the crane. In order to implement an IMU configuration into the simulation, one must first investigate how the IMU works.

There are lots of different IMU chips on the market, whereas some are more accurate, some more weather-proof and some vary in prices. The main idea behind this project is to investigate a cheap, easy accessible solution, whereas IMU chips may easily be replaced if broken, or added to several loads without a noticeable financial punishment. Therefore a cheap IMU is preferred. The main difference between a fairly cheap IMU chip and that of a more expensive version, is that a cheap IMU may produce more noise in a measurement as well as a more bias or drift data. For very exact measurements, the price for the IMU alone will render this solution financially unreasonable, thus by allowing some level of uncertainty, an IMU reliant system becomes more realistic.

In order to investigate how reliant the cheap IMUs are, a MPU6050 IMU chip has been used to sample gyroscopic and acceleration data. By connecting the MPU6050 chip to an Arduino UNO microcontroller using the built in I2C communication pins, the Arduino is able to read sample values of the different measurements. By further installing the MATLAB Support Package for Arduino Hardware, a custom script can be created in order to read the I2C data from the Arduino and into a Matlab variable. This yields a fairly stable and accurate way of reading live data from an IMU into a Matlab script, whereas further investigations of the IMU and its data can be done.

By leaving the IMU still on a levelled surface, it is possible to observe any bias or drift data. As the IMU is kept at rest, the data should represent zero movement, however if it yields a constant anomaly we have a bias added to the data. After running several long lasting zero-acceleration-tests on the given MPU6050 IMU chip, a certain but almost insignificant bias is present on the accelerometer data, as seen in Table 3. This bias is however quite small, and somewhat insignificant when working with offshore loads, whereas a single empty 10ft shipping container weighs approximately $1,300\ kg$, yielding a force calculated by equation 4, of $F = 12.748.71\ N$. By adding the bias to the calculation the force on the container is $F = 12.836.07\ N$, yielding a difference of $87.36\ N$ which is nearly insignificant compared to the $12,000+\ N$ Newtons of force naturally acting on the container. It should however be implemented a correction step in the model, to ensure the most reliable data possible for the control system.

$$F_z = m \cdot a_z = m \cdot g \tag{4}$$

Table 3: MPU6050 accelerometer bias test.

Axis	$\frac{m}{s^2}$	g	Anomaly Calculations	Bias
Ax	0.2668	0.0272	$0.2668 - 0$	$0.2668\ \frac{m}{s^2}$
Ay	0.5405	0.0551	$0.5405 - 0$	$0.5405\ \frac{m}{s^2}$
Az	9.8739	1.0069	$9.8739 - 9.8067$	$0.0672\ \frac{m}{s^2}$

3 Simulation

3.1 Simulation Goal

In order to advance on the research done in the specialization project, the Simulink simulation model has to be further improved. As the goal is to achieve an active load control to counter oscillations on the suspended load created by external disturbances, the crane must be able to move and react to external forces without the manual input. This can be achieved by implementing a new and more advanced control structure for the crane. Therefore three main goals has been decided such that the simulation will both control the crane based on the movement of the suspended load, and so the simulation reflects reality to some degree. These main goals are as described below.

The crane simulation was originally built with a set of manual inputs, as seen in Figure 9, whereas one input was controlling one specific joint of the crane, thus resulting in a fully manual crane where the operator has to use his intuition to counter disturbances. As this crane is supposed to become semi automatic, one of the main goals for the simulation is to implement a system whereas the user can input a set of orders for the crane rather than the velocity set points for the individual joints, and then allowing the crane to configure itself in accordance to said orders.

Furthermore, in order to control the oscillations of the suspended load, the crane must know how the load is moving in regards to the tip of the crane, hence referred to as the end effector or EE. Thereby the second main goal of the simulation will be implementing a simulated inertial measurement unit, or IMU, which would imitate that of a physical device.

When the simulated crane is able react to given commands, and receive data for how the suspended load is moving in regards to the end effector, one can start applying the movement data into the control structure of the crane. This way the crane will receive orders for how to move based upon the movement of the load, thus leading to a control structure for controlling the suspended load or cargo. This would be the third and final main goal of the simulation.

The simulation built and described in the subsequent sections will mainly become a working proof of concept, rather than a fully functioning control system. There will therefore be some simplifications, such as a limited number of degrees of freedom for the load oscillations. The simulation will to some extent be built in a modular fashion such that subsystems can be replaced or added with little effort, should one want to either make it more realistic in terms of degrees of freedom, or to add several data input sources.

3.2 Automatic crane configuration

As the end goal is to achieve a semi automatic crane, whereas the crane may move or correct its movement based on the external forces acted on the suspended load, the crane must be able to interpret the data from a sensor and then execute a necessary movement. In order to achieve this, the crane will be modulated as a robotic arm with three revolting joints, as seen in Figure 4b, and a system must be implemented to independently control the movement of the crane.

Throughout the earlier simulations, all movements of the crane was done through individually manipulating each robotic joint, or actuator. This rendered the crane to move, but each input only applies an individual change of speed in either the tower rotation, the main boom movement or the outer boom movement. In order to achieve a more sufficient automatic control, the base control structure has been changed from individual joint control, to a velocity speed control of the end effector.

By redefining the three different manual inputs, such that there is one for each of the axis in the global reference frame, one will yield a velocity input whereas each input accounts for the movement of the end effector along the respective axis. These manual inputs will collectively be referred to as $V_{EE\_desired}$, and defined as seen below. From earlier we had the input vector $\dot{q}$ as the manual input, which will now be calculated by the control structure.

$$V_{EE\_desired} = \begin{bmatrix} V_{EE\_x} \\ V_{EE\_y} \\ V_{EE\_z} \end{bmatrix}, \ \& \ \dot{q} = \begin{bmatrix} \omega_{tower} \\ \omega_{main} \\ \omega_{outer} \end{bmatrix}$$

This brings up the next step of the control structure, whereas the $\dot{q}$ vector will have to be calculated in order to move the EE in the desired directional speed. In accordance to the theory described by L. Sciavicco and B. Siciliano (*2*), one must first acquire the Jacobian matrix of the current crane configuration. This Jacobian can then be used to calculate the individual speeds of the joints, based on the desired velocity of the end effector (*3*), as seen in equation 5. In the field of robotics, the Jacobian is defined as a matrix which provides the relation between the joint velocities $\dot{q}$ and the end-effector velocities V_{EE} of a robot manipulator (*4*). Mathematically, the Jacobian is defined as a $m \times 6$ matrix, where the m columns correlate to m joints in the robot, and each row is a relation to either a axis velocity or a rotational speed, as seen below.

$$\dot{q} = J^{-1} \cdot V_{EE\_desired} \tag{5}$$

$$J = \begin{bmatrix} J_{11} & \cdot & \cdot & \cdot & J_{1m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ J_{61} & \cdot & \cdot & \cdot & J_{6m} \end{bmatrix}, \text{ where: } \begin{matrix} J_{1x} \rightarrow \omega_x \\ J_{2x} \rightarrow \omega_y \\ J_{3x} \rightarrow \omega_z \\ J_{4x} \rightarrow V_x \\ J_{5x} \rightarrow V_y \\ J_{6x} \rightarrow V_z \end{matrix}$$

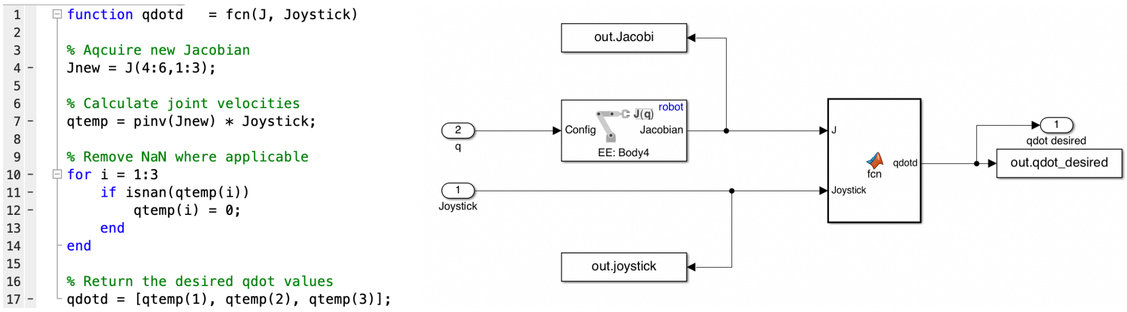
In the simulation of the crane, the robotic system is made up of five separate joints in order to fully mimic the crane, wire and the winch mechanism. However there are only three joints that will be autonomously controlled in this task. These joints are the rotation joint for the tower, the main boom, and the outer boom. The prismatic joint for the winch will be separately controlled, such that the winch will work independently of the rest of the crane. The final joint does not contain any actuation, thus moving freely, and only exists to mimic the movement of the wire at the suspension point. This means that the Jacobian used for calculations in the simulation will be a 3×6 matrix.

Furthermore, as the end effector speed $V_{EE\_desired}$ is given from the manual input, one will then need to narrow down the Jacobian further to only extract the necessary information for calculating the speed of the individual joint speeds. This has been done by ignoring the first three rows, such that the final Jacobian used for the simulation is a 3×3 matrix, as seen in equation 6.

$$\dot{q} = J_{new}^{-1} \cdot V_{EE\_desired} \tag{6}$$

$$\dot{q} = \begin{bmatrix} J_{41} & J_{42} & J_{43} \\ J_{51} & J_{52} & J_{53} \\ J_{61} & J_{62} & J_{63} \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_{EE\_x} \\ V_{EE\_y} \\ V_{EE\_z} \end{bmatrix}$$

By going back to the simulink work file, this calculation can be implemented by using two simulink blocks. The first block will be a ”Get Jacobian” block from the Robotics Toolbox (5). This block will yield the Jacobian matrix given the current joint configuration. This Jacobian will be a full 5×6 Jacobian matrix, due to the two extra joints for the wire. A matlab function block is then added, whereas both the changes to the Jacobian in accordance to the J_{new} seen above, and the $\dot{q}$ calculations will be conducted. The matlab function code, and how the blocks are connected can be seen in Figure 10. The q_{dotd} output from the matlab function block is now a 1×3 matrix with the desired velocities for each actuated joint such that the end effector velocity reaches it’s desired direction and magnitude.



```
function qdotd   = fcn(J, Joystick)

% Aqcuire new Jacobian
Jnew = J(4:6,1:3);

% Calculate joint velocities
qtemp = pinv(Jnew) * Joystick;

% Remove NaN where applicable
for i = 1:3
    if isnan(qtemp(i))
        qtemp(i) = 0;
    end
end

% Return the desired qdot values
qdotd = [qtemp(1), qtemp(2), qtemp(3)];
```

(a) Matlab function code. (b) Simulink representation.

Figure 10: Calculation of $\dot{q}$.

3.3 Load Control

The next step is to implement a system to stabilize and control the suspended load. For a regular, manually driven crane, this is done by moving the crane such that the movement counteracts that of the oscillating load. Here they crane operator observes how the load is moving, and then makes a decision made from this observation. For the automatically controlled load stabilizer, the system is based upon the same principle as the manually controlled one, whereas the system will observe the load oscillations, and then render the crane to move in a counteractive way. In order to see if the control system is working when it's implemented, a perfect measurement of the load movement will be collected from the simulation, and used as a reference for the control system. This perfect measurement will later be traded for simulated IMU data, with a more realistic take on bias, drift and noise.

This control system can be adapted to work in several different scenarios, however, in this project the scenario is that of where the crane is not receiving instructions to move, but the environmental disturbances yields a movement to the load. Meaning that the stabilization system will be turned off when the crane receives a manual input from a crane operator, and then activates when the load is supposed to be at rest.

Firstly, a case based system will be introduced to the simulink diagram, as seen in Figure 11. Here the total sum of the manual input joystick is calculated as in equation 7, and the solution u is then evaluated as shown in equation 8 in order to turn on or off the control system. This way the control system won't be counteracting the movement imposed by the operator, but instead stabilizes the suspended load as soon as the crane operator stops the crane.

$$u = \sum_{i=1}^{3} V_{EE\_desired}(i) \tag{7}$$

$$\text{Control Status:} \begin{array}{l} u = 0 : \text{ ON} \\ u \neq 0 : \text{ OFF} \end{array} \tag{8}$$

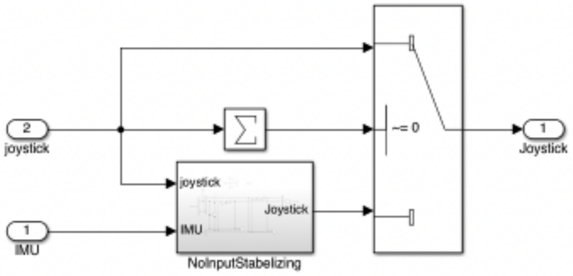


Figure 11: Input switch case.

As seen in the figure above, the control system is then implemented in a separate subsystem. This has been done in order to keep the systems apart, and to keep the simulink work file structured. One can see that the control system receives both IMU data, which now are perfect measurements, and the joystick signal.

The joystick data has been set as a input in order to easily scope the response of the system, however it can be used in a later stage if one chooses to keep the control system turned on even when the crane is being manipulated by an operator. As one can see in Figure 12, the IMU data, or more specific, the gyroscope data, is what's being used to counteract the oscillating load. By placing a gyroscope on the, or as close to the load as possible, one can observe the oscillating movement of the load as it's moving around the suspension point.

This gyroscope signal will then be integrated up to a position parameter, whereas position 0 is the load suspended straight down. The integrated signal is then scaled through a gain block, yielding a straight P-regulator for the position of the load. Whenever the disturbances, being wind, waves etc., are changing the position of the load, the system will then use the proportional signal to move the crane in a way to counter the oscillations.

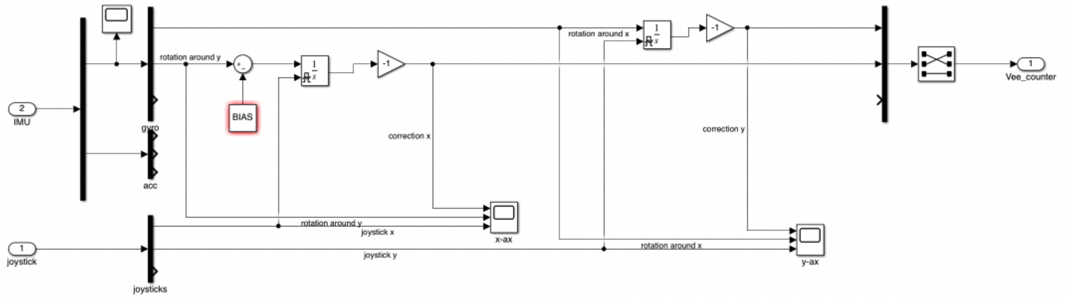
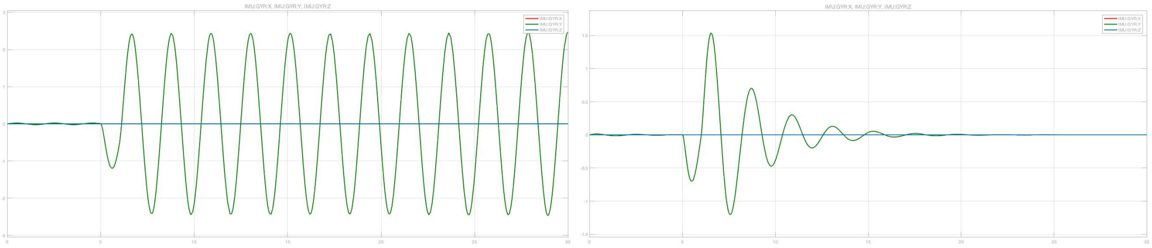


Figure 12: Anti oscillating control system.

In order to clearly see any differences in the response with and without a control structure, all natural damping of the load oscillations has been turned off, yielding a continuous oscillating motion after the load receives an impulse disturbance at 5 seconds. This can be seen in Figure 13a, whereas the green graph represents the movement around the Y-axis. When the system is activated, and proceeds to counteract the movement, one can see that the load becomes stationary again after a few seconds, as seen in Figure 13b. One can also observe that there is only movement around the Y-axis, this is due to the simulation being built with only one degree of freedom in the load suspension point. The reasoning behind this is to reduce the complexity of the simulation in order to clearly see if the system is working.



(a) No control system.

(b) Controlled load movement.

Figure 13: Comparison between with and without a control system.

3.4 Simulated IMU

As the previous simulations has proven that with the correct measurements, the crane would be able to stabilize a suspended load. However in a more realistic setting, the measurements from the load would not be as accurate due to most IMUs generate noise, bias and drift. From earlier we looked into the cheap MPU6050 IMU chip, and we saw a small bias and drift in the accelerometer measurements. Similarly the gyroscope measurements yields inaccurate data, as seen in Figure 14, where the gyroscope was held at rest for 60 minutes. After running the gyroscope test for 3600 seconds, one can see that there seem to be a small amount of drift, however the bias and noise seem to be the worst issue in regards to a control system.

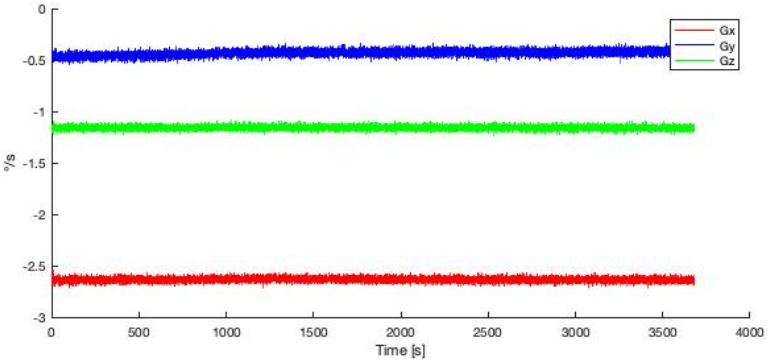
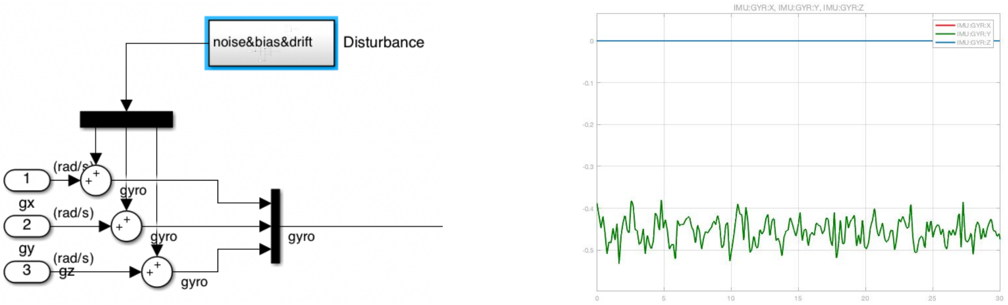


Figure 14: One hour static gyroscope test.

The control system is based upon an integrated signal from the gyroscope, and as such it is somewhat sensitive to noise (*6*). Furthermore, as the control is based upon the integrated angular velocity, the bias of the signal will lead the crane to believe the load is suspended at an angle that has to be corrected. The crane will then start moving to correct this angle, resulting in a constantly moving end effector.

In order to to incorporate a more realistic IMU measurement into the simulation, the data from the static test was loaded into the simulation as a time-series. This time-series was then layered on top of the perfect measurements from the simulation, yielding a noisy signal with both a natural drift and bias. The implementation and the realistic gyro signal can be seen in Figure 15.



(a) Simulink implementation.

(b) Realistic IMU measurement.

Figure 15: Implementation of realistic noise.

Now that the IMU signal contains drift, bias and noise, some extra steps must be taken in order to get a reliable control structure. A moving average calculation of the signal could solve most problems with both bias and drift, however this does not seem yield a high impact on the amount of noise in the measurements. For this the best solution would be to implement a filter to further reduce and remove any noise fluctuations. The end result from the moving average will then be a offset value that can be subtracted from the gyroscope signal before the integration, such that it counteracts the bias and drift from the measurement.

In the simulation the moving average uses thirty time steps to calculate the average value, this led to a more smooth bias and drift signal, as seen in Figure 16. As it uses previous time steps in its calculations, the average starts out with lots of fluctuations, however it quickly becomes more smooth as soon as enough data points have been collected.

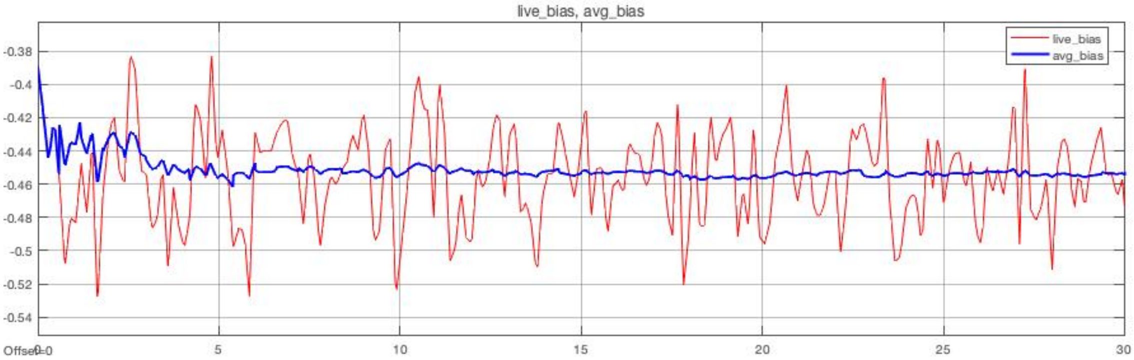


Figure 16: Moving average of bias vs. live bias.

This signal can now be used to counter the bias and drift from the IMU, yielding a signal that is centered around zero angular velocity when the load is suspended straight down. There is however some noise left on the signal, which may lead to a an integral windup during the transition between angular velocity to angular position. At this point a choice was made not to start looking into filtering of the signal, as the system seemed to be able to handle some noise. One can experience some integral windup during large changes in the manual input. In order to cope with this, an external integrator reset port was added to the integrator block (*7*), whereas the joystick input triggers the integrator block to reset it's built up error signal, when the joystick input returns to zero.

The control system can now both deal with somewhat realistic IMU readings and stabilize the load when external forces are acting upon it, given gyroscope measurements from the load, and positional and velocity readings from the actuators on the crane. Although this simulation is created with one degree of freedom in the suspension point, one can fairly easy exchange the joint configuration to add several degrees of freedom, thus resulting in a more realistic simulation in the future.

4 Sintef Crane System

4.1 Modeling and Simulation

One of the aspects for the thesis was to look at the new crane system acquired by Sintef, and look at the possibilities to further improve the crane in accordance to what have been discussed in the past sections. There is however a quite significant change in design between the boom and knuckle crane described in the Section 2.2.

First and foremost, the Sintef crane consists of only two revolute joints, and a prismatic part for extra reach, as seen in Figure 17. This yields a somewhat different model for simulations and crane control systems, and therefore a new DH table has to be made. The new DH-parameters can be seen in Table 4. Comparing the new DH-parameters to those given in Table 1, one can see that the new parameters include a second zero-length link for proper reference frame rotation, and the final movable joint has been swapped with a prismatic joint.

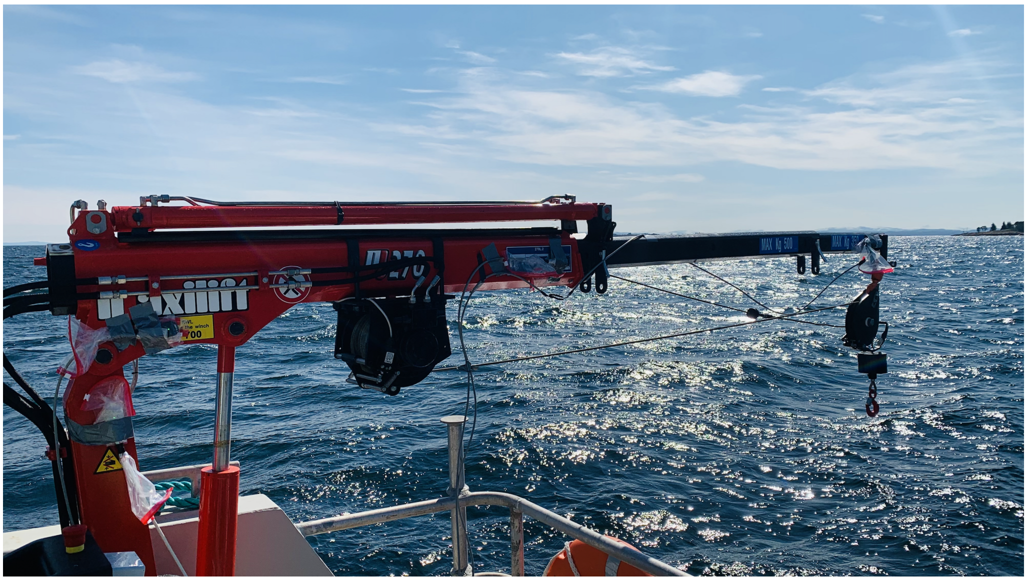


Figure 17: Sintef Crane

Table 4: DH-table for the Sintef crane

Link	α_{i-1}	a_{i-1}	d_i	θ_i	Joint
1	0	0	0.02 m	θ_1	Revolute
2	0	0	1.40 m	0	Fixed
3	$\frac{\pi}{2}$	0	0	θ_3	Revolute
4	$\frac{\pi}{2}$	0.80 m	0	$\frac{\pi}{2}$	Fixed
5	$\frac{\pi}{2}$	0	d_5	0	Prismatic

Using equation 2 for the A-matrices yields the following A_i^{i-1} matrices, which then results in the T_5^0 matrix seen in equation 9.

$$A_1^0 = \begin{bmatrix} cos(\varphi_1) & -sin(\varphi_1) & 0 & 0 \\ sin(\varphi_1) & cos(\varphi_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} cos(\varphi_3) & 0 & sin(\varphi_3) & 0 \\ sin(\varphi_3) & 0 & -cos(\varphi_3) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & a_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^0 = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \tag{9}$$

$$T_5^0 = \begin{bmatrix} sin(\varphi_1+\varphi_3) & cos(\varphi_1+\varphi_3) & 0 & d_5 \cdot cos(\varphi_1+\varphi_3) \\ -cos(\varphi_1+\varphi_3) & sin(\varphi_1+\varphi_3) & 0 & d_5 \cdot sin(\varphi_1+\varphi_3) \\ 0 & 0 & 1 & a_4+d_1+d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By following the same pattern as in section 2, this system will then be implemented into simulink through the use of URDF coding, and with the help of the robotics toolbox for Matlab. The simulink model can be seen in Figure 18. In this model there has been added two more prismatic joints in the same direction as the first one. This in order to more accurately simulate the movement of the retracting boom, which has several sliding parts. All of these prismatic joints will be actuated equally at the same time, resulting in the joints working together as one in order to extend the boom.

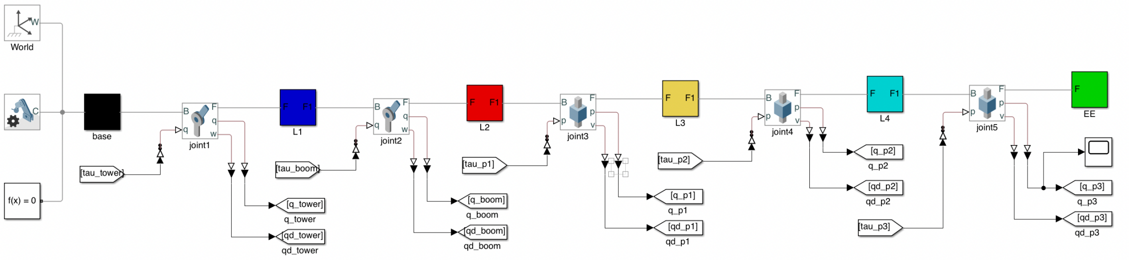
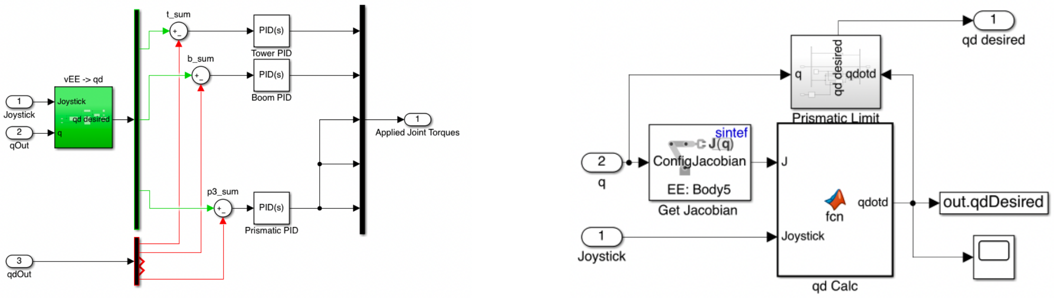


Figure 18: Crane model.

At this point, the simulation consists of an model of the crane, without any actuation on the joints. In order to control the crane, the control structure from the Boom and Knuckle crane needs to be fitted to work with the Sintef crane which has a different joint configuration.

The first thing that has to be solved is the changes to the Jacobian matrix for the new crane. As this crane does not yet have any suspended load, the Jacobian matrix will still be a 5×6 matrix. However the end effector will now be the outermost link, and thus the values from the fifth column is needed to correctly configure the crane. As seen in Figure 19a, the final three joints will receive the same velocity signal in order for the entire prismatic element to move as one. One can also see that the first two joints are receiving individual velocity signals.



(a) Sintef control structure.

(b) Calculation of $\dot{q}_d$.

Figure 19: Implementation of control structure for Sintef crane.

From this we can see that the Jacobian used for the calculations of $\dot{q}_{desired}$ requres the parameters for the joints $\begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$, thus resulting in the final Jacobian as seen in equation 10. Using this Jacobian, one can find the desired velocities for the individual joints, using the Matlab function block seen in Figure 19b. Now the simulated crane will be controllable using a manual input of desired velocities in the x-, y- and z-axis of the global reference frame. The simulation will then output a three dimensional model of the crane, as seen in Figure 20. The colors of the links on the output is a reference to the individually colored links in the simulink model seen in Figure 18.

$$\dot{q} = \begin{bmatrix} J_{41} & J_{42} & J_{45} \\ J_{51} & J_{52} & J_{55} \\ J_{61} & J_{62} & J_{65} \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_{EE\_x} \\ V_{EE\_y} \\ V_{EE\_z} \end{bmatrix} \tag{10}$$

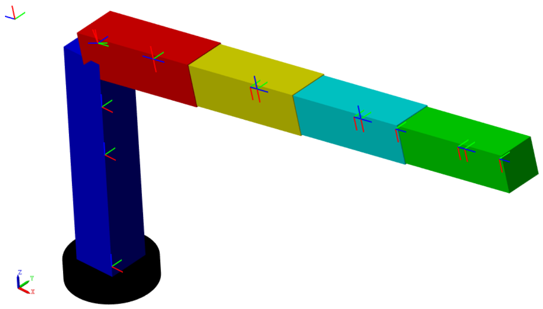


Figure 20: 3D simulated model of Sintef crane.

4.2 IMU measurements

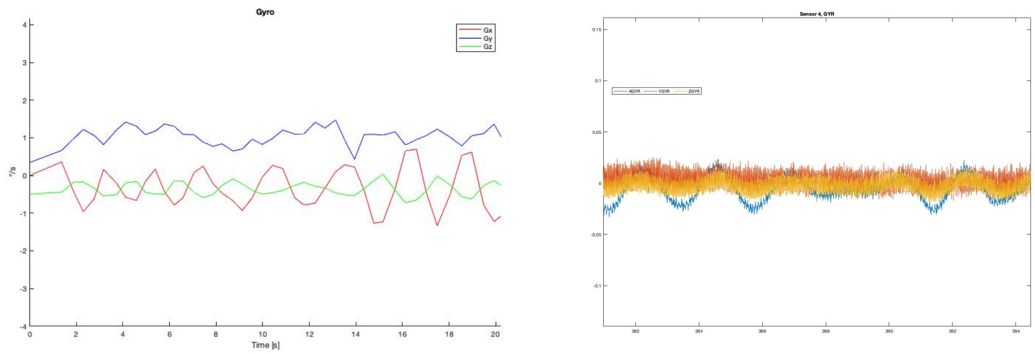
In order to evaluate the use of cheap IMU chips, such that the MPU6050, a series of measurements were conducted on the Sintef crane. The crane was already mounted on a boat, yielding it a perfect place to investigate how the IMU handles real environmental disturbances. The measurements were conducted both while the boat was docked and fairly stable, and in open waters, such that one can see how the IMUs react to different dynamic movements.

For a better understanding of how cheap IMUs handles dynamic disturbances, and to see how well they work in a realistic situation, a set of high end IMUs were attached to the same positions on the crane as the cheap ones. By comparing the sampled data from the different types of IMUs, one can investigate the differences between high end and low end units.

The IMUs for this test were located on the middle part of the boom, just short of the prismatic joint, which yields clear rotational velocity readings for all possible rotational motions the crane may endure. However, as the high end IMU at the tip of the crane did not save the recorded data, we do not have any readings of the prismatic movement.

In Figure 21 one can see the difference between a low end and a high end IMU when the boat is docked and at rest. We can observe that there's less bias on the more expensive IMU, and that the MPU6050 readings have far less data points. The motions readings on the x-axis does seem to be correlating, yielding a similar signal for the wave motion endured while docked. Note that the colors of the axis are different on the high end and low end IMUs. The axis color descriptions can be seen below.

$$\text{Low end: } \begin{cases} x\text{-axis} = \text{Red} \\ y\text{-axis} = \text{Blue} \\ z\text{-axis} = \text{Green} \end{cases} \quad \& \quad \text{High end: } \begin{cases} x\text{-axis} = \text{Blue} \\ y\text{-axis} = \text{Red} \\ z\text{-axis} = \text{Yellow} \end{cases}$$

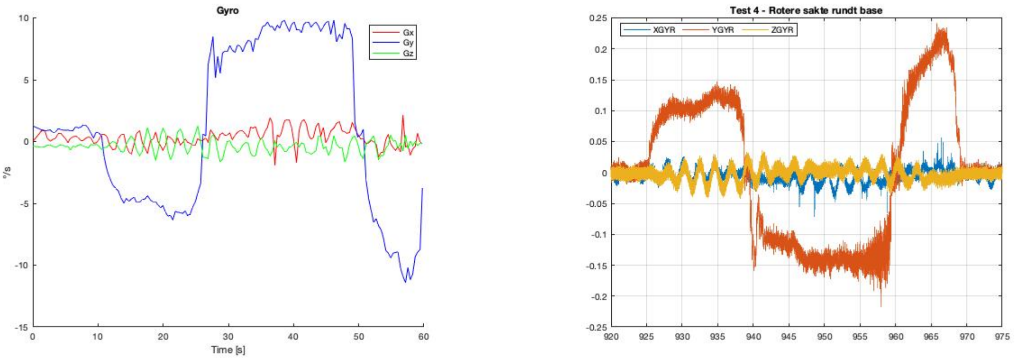


(a) Low end IMU.

(b) High end IMU.

Figure 21: Docked test with no crane movement.

Figure 22 is representing a measurement whereas the boat was docked, and the crane was slowly rotated 90° to the left, then slowly 180° to the right, and finally 90° to the left such that the crane ended up in its original configuration. Although the IMUs had a opposite definition of positive and negative readings from the axis, one can see here that the recorded motions are a close match, and yields a clear representation of the crane operation. As the boat was docked and the waves were fairly small, there is a vast difference between the crane movement signal and the dynamic disturbances from the ocean.

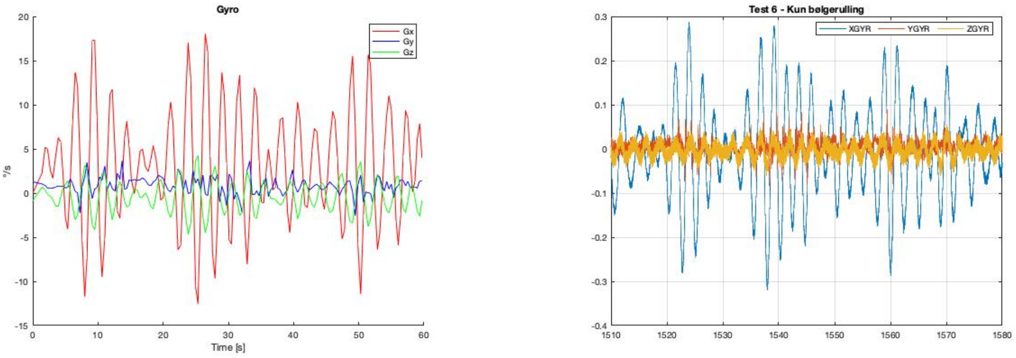


(a) Low end IMU.

(b) High end IMU.

Figure 22: Docked test with crane movement.

Once the boat got into open waters, one can see way more disturbance on the measurements. Figure 23 shows a measurement where the crane was at standstill, whereas all the movements recorded by the IMUs were due to the waves acting upon the boat. One can see that the waves have a high impact on both of the IMUs, where both are yielding a fairly precise reading of how the waves are rocking the boat. Disregarding the differences in bias and sample times, both the high end and low end IMU are extracting signals that looks to be almost identical.



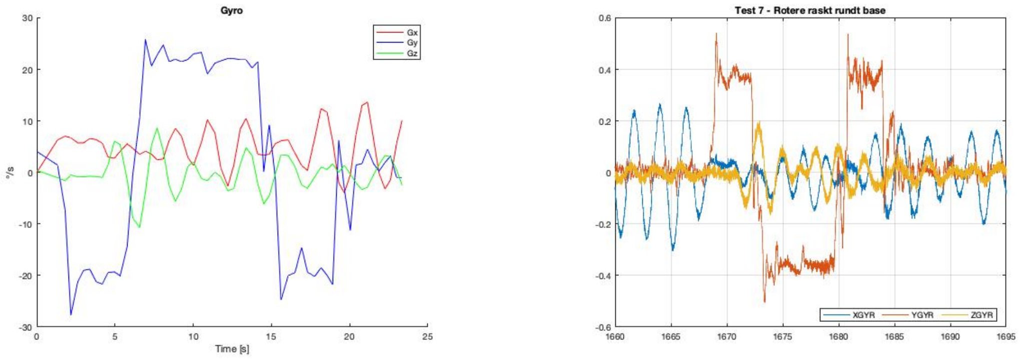
(a) Low end IMU.

(b) High end IMU.

Figure 23: Open waters test with no crane movement.

The final test was a recreation of the test in Figure 22, with regards to the crane movement. The test was now conducted on open waters in order to see how the movement would be measured while exposed to a significantly higher level of disturbance. From Figure 24 we can observe that even with a lot of disturbance, one can still clearly see both sensors yields a clear signal for how the crane is moving.

One interesting observation is how the wave motion transfers between the x- and y-axis during the 90° crane turns. As the crane turns, so does the individual frame of reference for the IMU, which leads the waves to be measured by a different axis. Although this is as expected, it may impose problems if a crane were to be controlled by IMU measurements instead of actuator measurements.



(a) Low end IMU.

(b) High end IMU.

Figure 24: Open waters test with crane movement.

5 Discussion

Looking at the results of the primary simulation, the resultant control structure seem to be working as intended. As a disturbance acts upon the suspended load, the crane automatically counters the movement of the disturbance, and regaining control of the load. It is intriguing to see how small movements in the crane may have such a large impact on the suspended load, and how well the control structure copes with the natural noise, bias and drift from the IMUs. Although this seem to be working well in the simulations, there are several sources of error that may lead to a malfunction when the control structure is transferred to a real system.

Firstly, the crane simulation only allows the load to oscillate in a single degree of freedom. In a real scenario the load will be able to swing in three dimensions. The system may encounter problems in dealing with oscillations in different directions, however by correctly breaking down the oscillations, simple crane motions in the x- and y-axis should be able to significantly reduce and stabilize the load.

Furthermore, there has not been implemented any regulations for movement in the z-axis. By going into the project, the assumption was that a operational and adequate heave compensation was already in place, rendering an extra system to regulate the motion in the z-axis a redundancy. By measuring the length of the wire between the suspension point and the load, one could however implement a solution that takes the rotation data and the length of the wire to calculate heave motions. This could then be implemented into the system in order to have one system that solves both the oscillation and heave problems.

One of the issues that has been raised during the project is the use of a measured angular position (q) and the angular velocity ($\dot{q}$) of the joints. The system takes into account that the measurements of these positions and velocities are existing and near perfect. This may not be the case in most cranes, however by knowing the startup configuration, one could use IMUs on the different beams and joints to measure the velocities. These velocities could then be integrated to gain the position (q) of the joints, compared to the start configuration. The IMU measurements would need some fine tuned filtering, in order to get the position data as accurate as possible in order for the control system to work as well as possible.

Although these issues may hinder a direct application of the control structure to a operational crane, through further investigations, and implementation of the listed workarounds, one should be able to fully stabilize the suspended load on a real crane. Given that some cranes already have a digital control system to control actuators and receive data from sensors, the main issue to address in the future is that of implementing the load control in three dimensions. The second issue would then be filtering of IMU data from the load, such that the control system receives a more smooth and readable signal.

These two future implementations should be enough to ensure the system working well enough on some digital Boom and Knuckle cranes with the same joint configuration.

One of the more interesting findings were that of the similarities between the cheap IMU chips and the more expensive ones. The main difference is the sample time, however this may be due to how the code for receiving the MPU6050 data was written. The code ran through a loop, whereas for each iteration the code grabbed the current value from 4 different IMUs placed around the crane. This may be the cause for the low sampling time for each IMU.

By either advancing the code to be running four different loops on four different computer threads for each IMU, or by using one computer for each IMU, one may be able to simultaneously read data from the IMUs, and thus increase the individual sampling rate for the cheap IMUs. This could possibly increase the sampling rate enough to compete with the high end IMUs whith regards to speed.

Furthermore, the cheaper IMUs seem to be more prone to add bias on the measurements. There was no drift test conducted on the high end IMUs, however they seem to be more stable in this department as well. The bias and drift does not necessarily pose as a major issue, given a sufficient way of neutralizing it. As seen in the simulation, by implementing a moving average system, one can minimise the impact of bias and drift, such that the signal can be used for further calculations with minimal error.

The main drawback to use IMUs are the amount of noise on the measurements. Although the load stabilizing system seem to be working in spite of the noise on the measurements, using IMUs to approximate the configurations of the crane may need filtering to remove as much noise as possible. Noise may lead to lots of integral windup and false configurations in the system, which would lead the system to try moving the crane into odd or even impossible positions.

Overall the use of cheap IMUs does not seem to impose too much problems for these tasks, given a few modifications to the simulation, and thus the financial benefits seem to outweigh the issues imposed by using cheap IMUs compared to more expensive ones. This does however hold for these systems, where the small errors can be disregarded. For systems that require a high level of precision, cheap IMUs may impose too much error and problems for the system to work as well as possible.

If one were to implement implement the simulated load control structure, with the proposed upgrades, one should be able to control the load using data from a cheap IMU. However using cheap IMUs to control the crane itself may be a bigger issue, as the control structure relies on near perfect data for how the crane is moving. With proper filtering, and a way to remove the wave oscillations from the crane measurements it may work. A way of removing the oceanic waves may be to keep a fixed IMU on the boat itself, and use the measurements from this to counter the wave measurements on the appropriate axis of the IMUs on the crane.

Bibliography

(*1*) A. J. Strand. 'Offshore Cranes'. In: *Project report at NTNU* (2020).

(*2*) L. Sciavicco and B. Siciliano. *Modeling and Control of Robot Manipulators*. Springer, 2000.

(*3*) P. Corke. *Inverting the Jacobian Matrix*. URL: https://robotacademy.net.au/lesson/inverting-the-jacobian-matrix/ (visited on 15th Mar. 2021).

(*4*) S. Krishna. *Jacobian*. URL: https://www.rosroboticslearning.com/jacobian (visited on 20th Mar. 2021).

(*5*) MathWorks. *Robotics Systems Toolbox*. URL: https://www.mathworks.com/products/robotics.html (visited on 15th Feb. 2021).

(*6*) K. J. Astrom and L. Rundqwist. 'Integral Windup and How to Avoid It'. In: *American Control Conference* (1989). DOI: 10.23919/ACC.1989.4790464.

(*7*) MathWorks. *Integrator*. URL: https://www.mathworks.com/help/simulink/slref/integrator.html?searchHighlight=integrator (visited on 2nd Apr. 2021).

Appendix

A List of External Files

This is a guide to where the different files may be found in the attached file structure. As there has been several variants of the simulations, as well as linked support files to ensure that these simulations are working, the placement of these files while running the simulations are important.

Appended_files.zip/

- init_crane.m *(Main Matlab file, run this to open simulation)*
- createRobotModel.m *(Matlab support file)*
- craneSim.urdf *(Crane model written in URDF)*
- craneSim.slx *(Simulink support file)*
- craneRBT.slx *(Simulink support file)*
- Crane.slx *(Simulink work file for crane simulation)*
- calculations.m *(Calculations of transform matrices)*
- arduinoTest.m *(Test I2C IMU data reception)*
- **SINTEF/**
  - initiate_sintef.m *(Main Matlab file for Sintef-crane)*
  - print_test_fig.m *(print IMU data as graphs)*
  - SINTEFcrane.slx *(Main Sintef Simulink file)*
  - SINTEFcrane.urdf *(URDF model for Sintef crane)*
  - sintefRBT.slx *(Simulink support file)*
  - **HighEnd/** *(Test files made by Sintef + data from high end IMU)*
  - **Measurements/**
    - SINTEF_IMU.m *(Rewritten I2C data reception code for offshore test)*
    - **testFiles/** *(IMU readings from offshore tests)*
- **Variables/** *(Stored values from 60m static IMU test. Used in simulation)*
- **Video/** *(Video of working crane stabilization simulation)*
- **Figures/** *(Graphs from several tests)*
- **Reference/** *(Specialization project report)*

NTNU
Norwegian University of
Science and Technology