# Offshore Cranes

## TPK4560 - Specialisation Project


*Figure 0: Cranes on a vessel.*[1]

Strand, Aslak J.

18 December 2020

[1] Strand, Aslak. derivative of "Zhen Hua 19." by Adam Jenkins, *Flickr*, 12 Apr. 2010, www.flickr.com/photos/37796451@N00/4526010078. Accessed 8 Dec. 2020. Licensed with CC BY 2.0. To view a copy of this license, visit https://creativecommons.org/licenses/by/2.0/.

[ page left blank intentionally ]

# Preface

As a former satellite engineering student, control theory and stabilisation is an interesting field of study. By combining my old field of interest with subsea and offshore technologies, the choice fell quickly at looking into control and stabilisation of offshore cranes. Offshore crane control spiked my interest as there is a lot more beneath the hood than what seems at first glance. The way everything is dynamically connected, from how the waves moves the load, and how the load will then rock the ship and thus creating more waves. It seem to be a never ending loop of disturbances which would be quite beneficial to reduce.

This text will serve as a preset to a bigger study, to which the end goal is to stabilise and control a hanging load which is subject to lots of disturbances. By taking into account the basics of cranes and how they operate, it's possible to predict a possible solution for how to control the load. In addition to the investigation, this text will show how to build a simulation of a crane with a load, and may therefore be used as a stepping stone for future simulations and investigations concerning the control it self during various disturbances. The text itself will consist of a mixture of theoretical, mathematical and practical descriptions about cranes, robotic manipulators and Inertial Measurement Units. Some illustration has been included to further the readers understanding.

I would like to take the opportunity to further extent my regards to my supervisors, Mr. Holden from NTNU and Mr. Ohrem from SINTEF, for all the theoretical and practical help they have offered. I would also like to thank Mr. Sollie from the Department of Engineering Cybernetics at NTNU for all help and theoretical inputs regarding IMU sensors.

*Aslak J. Strand*

# Table of Contents

# I. List of Figures

# II. List of Tables

# III. Nomenclature

| | |
|---|---|
| ARC | Adaptive Robust Controller |
| DH | Denavit–Hartenberg |
| 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. Literature Study

## 1.1. Background

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[2], 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[3], or smaller machinery such as the claw in a Claw Machine, as seen in Figure 1b[4], 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.

*Figure 1a: Tower Crane.*


*Figure 1b: Claw Machine.*

*Figure 1: Various crane designs.*

## 1.2. Problems With Offshore Cranes and Operations

As stated in subsection 1.1, 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[5], 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.

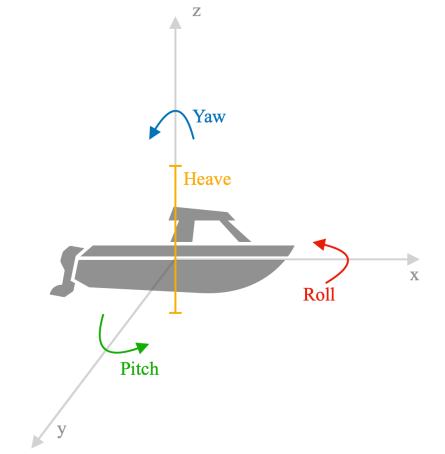


*Figure 2: Roll, pitch, yaw and heave.[6]*

In accordance with the Norwegian Standard ES-5516, cranes used for offshore operations can be categorised into three main groups, depending on placement and usage. The first category is offshore cranes, which is used to handle cargo in open seas, between fixed or floating structures and ships, whereas at least one structure or ship is in motion[7]. The second category is ship cranes, which are cranes aboard a ship, where the crane is used for loading and unloading of cargo at a port. The final category is a deck crane, which are fixed cranes for transport of cargo on deck and between the ship and supply vessels.

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[8].

There has been a lot of research and solutions made for compensation of the dynamic works, especially in heave translation, whereas a heave compensator will counteract some of the translation in the z-axis, from Figure 2, and thus reduce the chances in tension of the wire. However the additional rotations are somewhat more difficult to predict as they appear more random and thus harder to counteract. Therefore the purpose of this article is to investigate a possible solution to controlling the load in spite of random changes in roll, pitch and yaw of the vessel, assuming that a adequate heave compensator has been installed in order to reduce the changes in tension of the wire.

## 1.3. Possible Tracking and Control Structures

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[9]. 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 3, 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 4, 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.

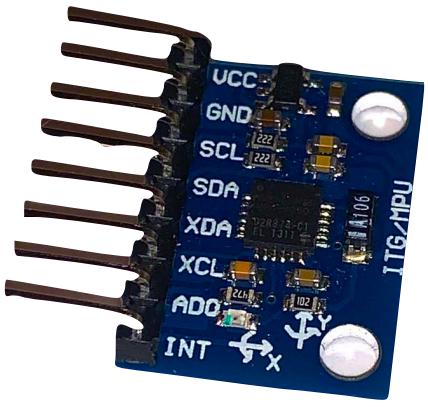


*Figure 3: MPU6050 IMU chip.[6]*

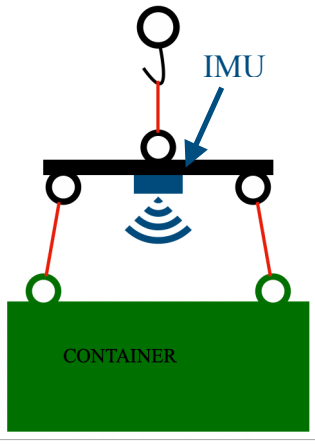


*Figure 4: IMU on lifting beam.[6]*

# 2. Modeling

## 2.1. Simple Crane modeling

For a functional implementation of the IMU data into the crane control system to be investigated, a simulation of the system is an effective way of designing the system and testing if the implementation works. Therefore a model of a crane will be needed for future simulations, and for the purpose of keeping the implementation as realistic and adaptive as possible, the crane model will be based on a Boom and Knuckle crane, which are frequently used for offshore craning operations.

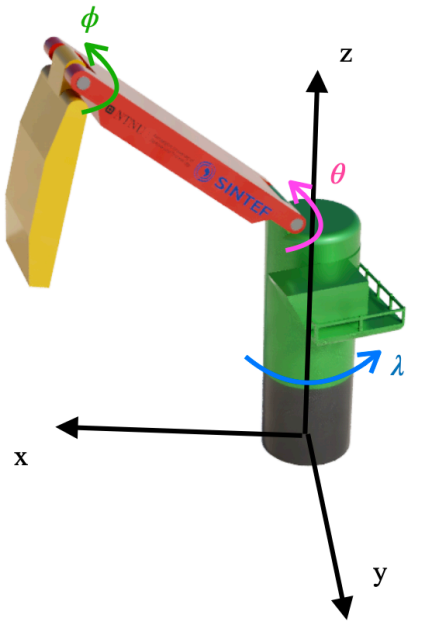


*Figure 5: Boom and Knuckle model.*[6]

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 5, 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 ($\lambda$), between tower and main boom ($\theta$) and between main and outer boom ($\phi$).

This design can then be split into DH-parameters[10], 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.
- $\alpha_i$: 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)
- $\varphi_i$: 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 6, whereas the cylindrical shapes illustrates revolving joints.

*Table 1: DH-parameters of model.*

| Link | $a_i$ | $\alpha_i$ | $d_i$ | $\varphi_i$ | Joint |
|---|---|---|---|---|---|
| $l_1$ | 0 | 0 | $d_1$ | $\varphi_1$ | *Revolute* |
| $l_2$ | 0 | $\frac{\pi}{2}$ | *0* | $\varphi_2$ | *Revolute* |
| $l_3$ | $a_3$ | *0* | *0* | $\varphi_3$ | *Revolute* |
| $l_4$ | $a_4$ | *0* | *0* | 0 | *Fixed* |

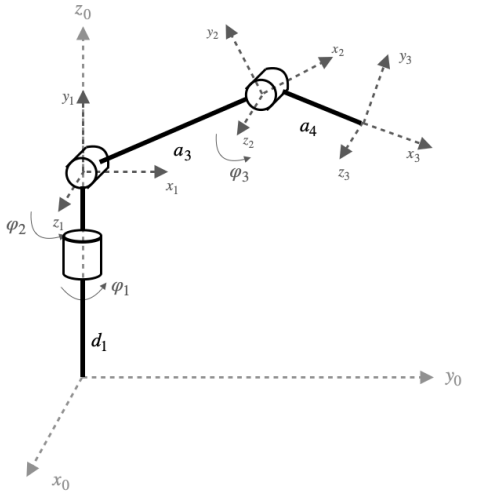


*Figure 6: Illustration of DH-parameters.*[6]

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 $A$-matrices for each joint can be calculated according to equation (2).

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

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

Using equation (2) for the links $l_1$ through $l_4$ yields the following $A$-matrices, and thus resulting in the $T$-matrix given in equation (3), which can be used in the simulation of the crane in order to yield correct movement between end effector and base.

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

$$A_2^1 = \begin{bmatrix} cos(\varphi_2) & 0 & sin(\varphi_2) & 0 \\ sin(\varphi_2) & 0 & -cos(\varphi_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} cos(\varphi_3) & -sin(\varphi_3) & 0 & a_3 cos(\varphi_3) \\ sin(\varphi_3) & cos(\varphi_3) & 0 & a_3 sin(\varphi_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\Downarrow$$

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

$$T_4^0(q) = \begin{bmatrix} cos(\varphi_1 + \varphi_2)\, cos(\varphi_3) & -cos(\varphi_1 + \varphi_2)\, sin(\varphi_3) & sin(\varphi_1 + \varphi_2) & cos(\varphi_1 + \varphi_2)\, cos(\varphi_3)\, (a_3 + a_4) \\ sin(\varphi_1 + \varphi_2)\, cos(\varphi_3) & -sin(\varphi_1 + \varphi_2)\, sin(\varphi_3) & -cos(\varphi_1 + \varphi_2) & sin(\varphi_1 + \varphi_2)\, cos(a_3)\, (a_3 + a_4) \\ sin(\varphi_3) & cos(\varphi_3) & 0 & d_1 + a_3 sin(\varphi_3) + a_4 sin(\varphi_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

## 2.2. Crane Modelling with Wire and Load

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. Then we are left with the spherical motion in the suspension point of the crane as the load starts swinging due to either crane movement or other disturbances.

This can be modelled as two revolving joints with a 90° $\alpha$-offset to each other, and with no distance between. This would result in a revolting joint that can rotate around two axis which together forms a spherical motion. 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.

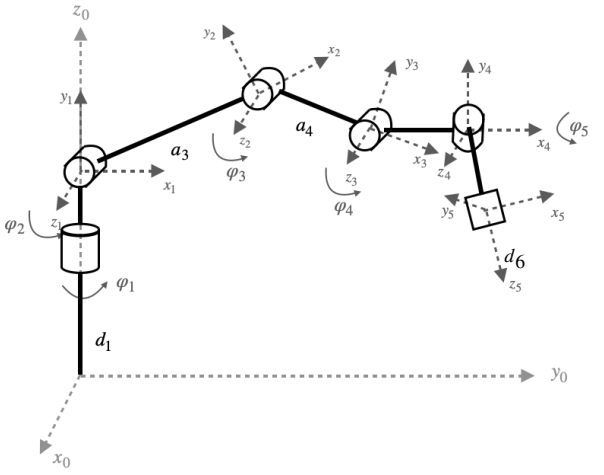


*Figure 7: Illustration of complex model.*[6]

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 7, whereas a square now represents the prismatic hosting joint, with it's extension axis along the prismatic z-axis.

*Table 2: DH-parameters of advanced model.*

| Link | $a_i$ | $\alpha_i$ | $d_i$ | $\varphi_i$ | Type |
|---|---|---|---|---|---|
| $l_1$ | 0 | 0 | $d_1$ | $\varphi_1$ | *Revolute* |
| $l_2$ | 0 | $\frac{\pi}{2}$ | *0* | $\varphi_2$ | *Revolute* |
| $l_3$ | $a_3$ | *0* | *0* | $\varphi_3$ | *Revolute* |
| $l_4$ | $a_4$ | *0* | *0* | $\varphi_4$ | *Revolute* |
| $l_5$ | *0* | $\frac{\pi}{2}$ | *0* | $\varphi_5$ | *Revolute* |
| $l_6$ | *0* | $-\frac{\pi}{2}$ | $d_6$ | 0 | *Prismatic* |

By using equation (2), the additional matrices $A_5^4$ and $A_6^5$ are computed as:

$$A_5^4 = \begin{bmatrix} cos(\varphi_5) & 0 & sin(\varphi_5) & 0 \\ sin(\varphi_5) & 0 & -cos(\varphi_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Furthermore, the rotation matrix will then be calculated as seen in equation (4).

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

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

# 3. Simulation

## 3.1. Simulink Implementation

By looking at the mathematical model of the crane, a more profound understanding of the movement of cranes has been established. Thus the method of implementing a crane model into MatLab and Simulink becomes easier. By choosing to implement the crane as a robotic arm manipulator opens up the ability to change and adapt the robotic arm at will, rendering it easier to add links and joint, as well as random movement to the base when needed.

There are various ways of implementing a robotic manipulator into Simulink, whereas the most intuitive way is by utilising the Simscape™ 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. These URDF files can layout the basic structure shapes and sizes, as well as the placement and rotation of the joints. The code builds on the principle that a robotic manipulator consists of parent and child components, where the parent component is link $l_i$ and the child is the link $l_{i+1}$, and thus placing the joint in between. A basic URDF sample code for a two-link arm can be seen in Figure 8.

```
<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 8: URDF sample code.[6]*

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 9a which then is displayed within the simulation as seen in Figure 9b.

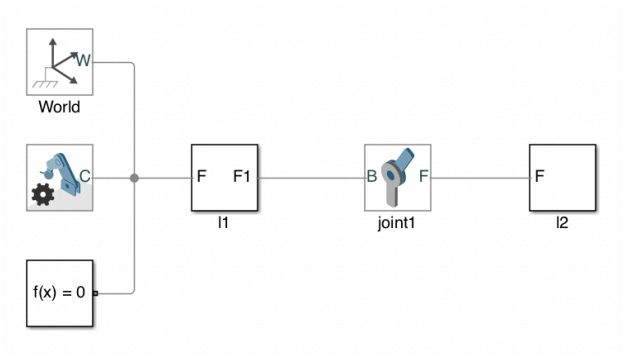


*Figure 9a: Simulink Model.*

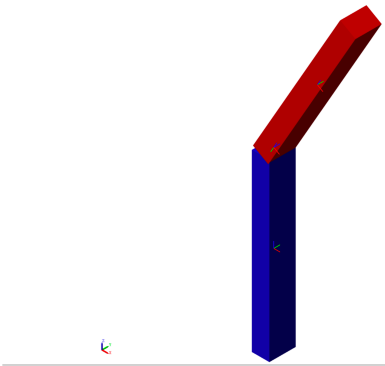

*Figure 9b: Simulated representation.*

*Figure 9: Two-link sample model.[6]*

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, as seen in the Appendix, 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 10. 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[11] in order to keep the movement as realistic as possible.

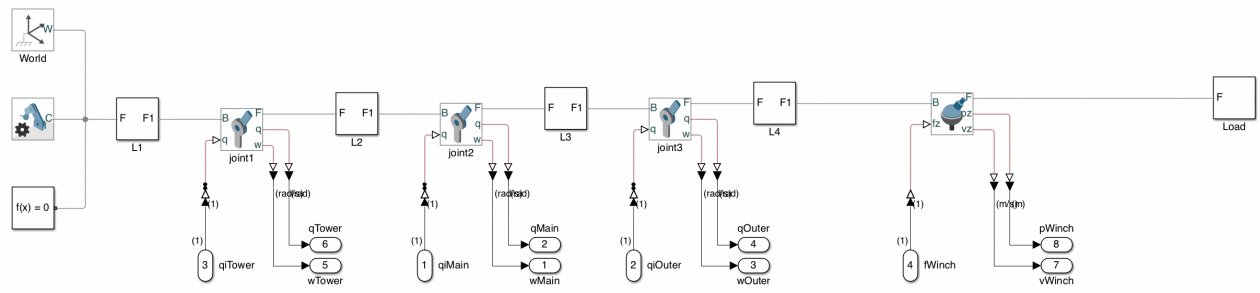


*Figure 10: Crane simulink model.[6]*

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 11.

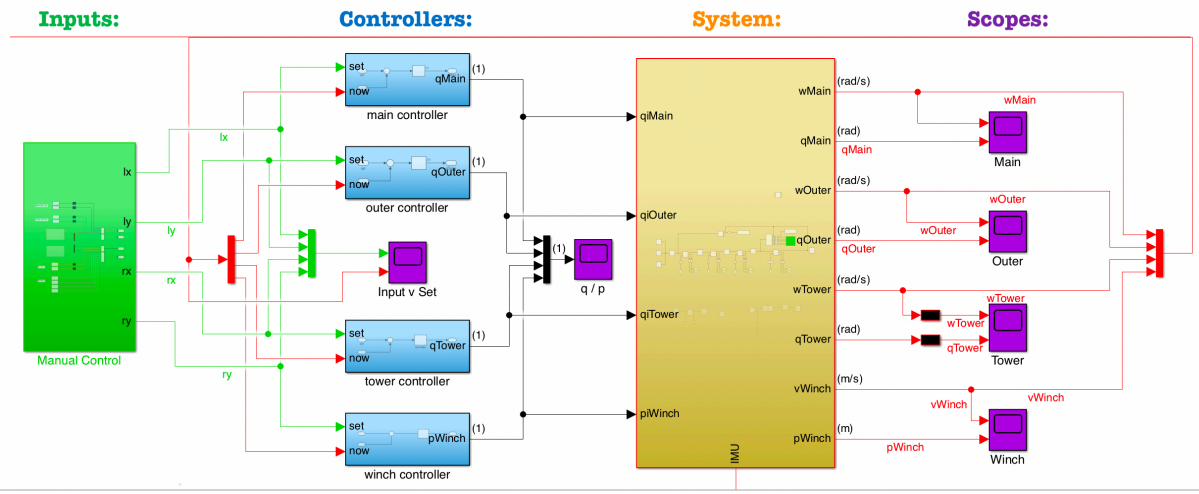


*Figure 11: Full simulink implementation.[6]*

This implementation results in an operative crane, whereas the tower rotation speed, angular velocity off both booms, and the winch speed can be manipulated by the joystick input. As a normal joystick has an x- and a y-axis with positive and negative values, and usually resets to the middle position when let go of, the setpoint was made to control the various velocities of the actuators. This results in a more accurate control, since the user may let go of the joystick for the crane to stand still in its current position. If the joystick were to control the angle or position of the joints, then the operator would need to hold the joystick in the position that represents the desired value.

At this point the crane moves freely with few limitations. There are a few limits to the joint rotations specified in the URDF, however these may easily be changed in the future within the Simulink joint blocks. Furthermore, the foundation of this crane is that fo a fixed structure, resulting in a steady crane simulation without outside disturbances. Future work may include adding another element at the bottom, which would imitate the movement of a boat at sea.

The Simscape environment used for the simulation does come with a 3D render, which leads to the final simulation being rendered and presented as a 3D model, as seen in Figure 12. By using the 3D render, you can easily see the movement of each joint as well as the effect these movements impose on the hanging load beneath the suspension point.

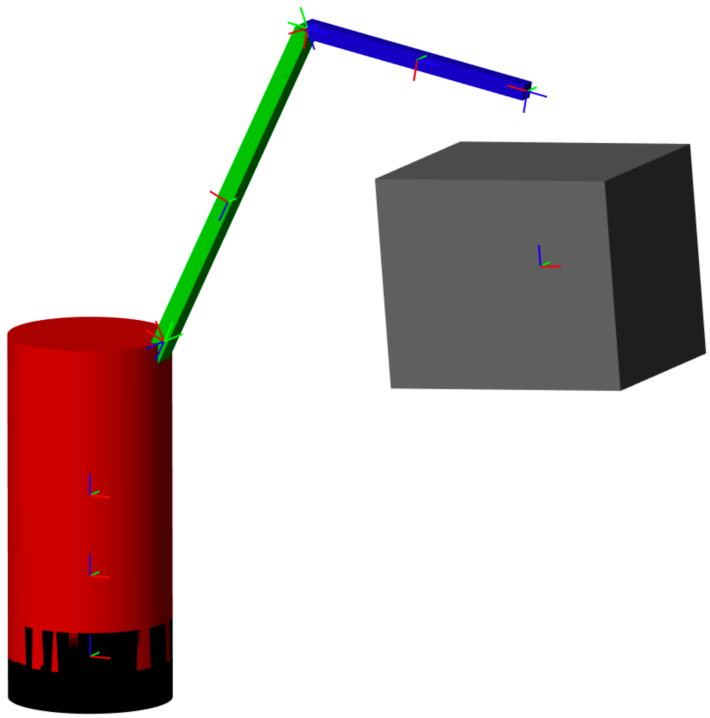

*Figure 12: 3D render of crane model during simulation.*[6]

## 3.2. 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. Bu further implementing the MATLAB Support Package for Arduino Hardware, a custom script, as seen in the Appendix, 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. The input data can then be plotted into graphs to illustrate how the acceleration and angular values change over time. A graph for acceleration can be seen in Figure 13a, and a graph for the angular data can be seen in Figure 13b. Note that the two graphs are from two separate tests, and the acceleration in Figure 13a does does not correlate to the angular velocities in Figure 13b.

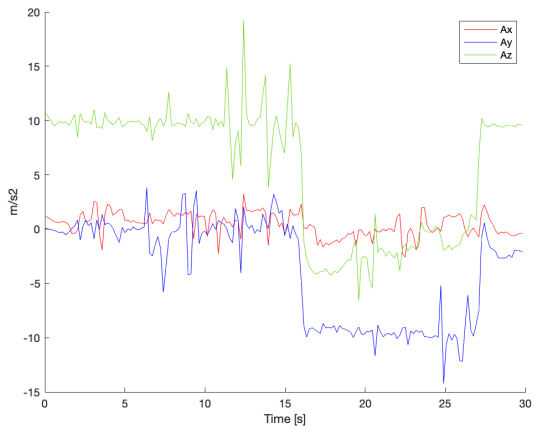


*Figure 13a: Acceleration data from IMU.*

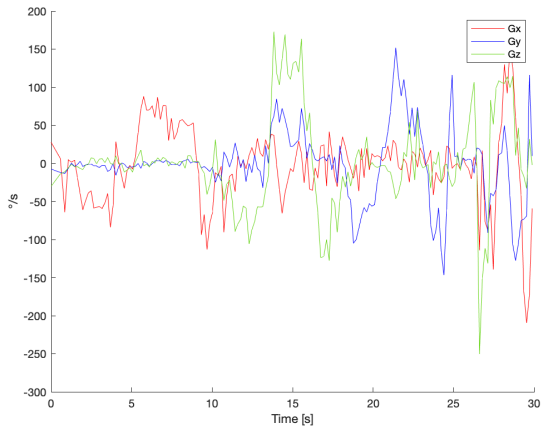


*Figure 13b: Angular velocity data from IMU.*

*Figure 13: Graphical presentation of IMU data.[6]*

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. Furthermore, if the anomaly increases or decreases at a certain rate, the data is drifting. These anomalies should be presented in the simulation such that the modelled system may counteract the faulty values, and yield a clean IMU signal.

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 the first ten seconds of Figure 13a, there is a constant acceleration in the upward along the z-axis of approximately 10 $\frac{m}{s^2}$, this is reasonable as the accelerometer is at standstill, and therefore it must be a force acting in the upwards direction to counter the constant gravitational pull downward at $9.81\ \frac{m}{s^2} = 1\ g$. By accumulating the accelerometer data and calculating the average, a bias can be 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 1,300.0 $kg$[11], yielding a force calculated by equation (5), 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+ 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.

$$F_z = m \cdot a_z = m \cdot g \tag{5}$$

*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 $\frac{m}{s^2}$ |

# 4. Results and Future Research

Theoretically it should be possible to utilise the data from an IMU on the load in order to keep the load stable during unpredictable disturbances such as wind and waves. There are however a few points that must be handled in future research in order to compute a definitive answer. As of now the simulation has not yet implemented IMU data for the load. This can be handled by using a transform sensor block in order to get the changes between the reference frame of the load compared to the reference frame of the suspension point. This transform data must then be subjected to a form of disturbance and bias similar to that of the working IMU. This would then simulate a working IMU on the hanging load, and could be further used through calculations and a Kalman filter to add input to the controllers. The mathematical models for splitting these data into the different joint controllers has not been computed, but should be finished during future research.

Furthermore, some extra research is needed on the given MPU6050 chip in order to find drift and bias for the gyroscope. From Figure 14 you can see that the chip at rest yields a varying signal for all gyroscope axis, which must be investigated further in order to find bias and drift data.

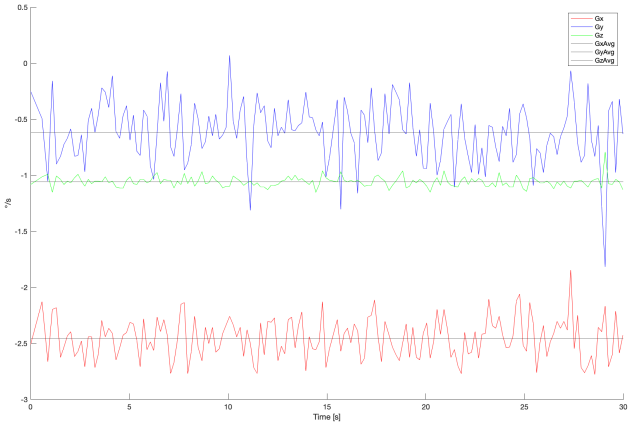


*Figure 14: MPU6050 gyroscope bias test.*[6]

Finally the model does not take into account any environmental disturbances on the load or the foundation. This is necessary in order to see if the crane is able to correct for the unpredictable movement of the load due to external forces and rotations. It does however result in a a simulation that adequately shows how the load moves due to the forces from the crane it self.

By further investigating the missing mathematical models and IMU data variations, and implementing the missing functions, the model should become sufficiently advanced in order to show if the load may be controlled using IMU data. In a fully functional system, redundancy is often necessary in order to operate safely and uninterrupted. By showing that the IMU data can be used by itself, it would absolutely be a working add-on to a more complex system with several types of measurement data, such as computer vision, GPS, and LIDAR. By incorporating several measurement methods, the end result should increase in accuracy, yielding a more reliable system.

# 5. Bibliography

[2] Hugsted, Reidar. “Kran – Anleggsteknikk.” *Store Norske Leksikon*, 11 July 2019, snl.no/kran_-_anleggsteknikk. Accessed 7 Dec. 2020.

[3] Bradhoc. “Tower Crane in the Fog.” *Flickr.com*, 23 Nov. 2011, www.flickr.com/photos/58719682@N07/6447382615. Accessed 7 Dec. 2020. Licensed with CC BY 2.0. To view a copy of this license, visit https://creativecommons.org/licenses/by/2.0/.

[4] Clark, Rusty. “Claw Machine at the Mall,” *Flickr.com*, 1 Nov. 2016, www.flickr.com/photos/23206546@N04/30260790683. Accessed 7 Dec. 2020. Licensed with CC BY 2.0. To view a copy of this license, visit https://creativecommons.org/licenses/by/2.0/.

[5] Oblack, Rachelle. “Why Is Wind Speed Slower over Land than over Ocean?” *ThoughtCo*, 13 Apr. 2019, www.thoughtco.com/wind-speed-slower-over-land-3444038. Accessed 23 Nov. 2020.

[6] Strand, Aslak. “Illustrative Content Created for Specialisation Project”
Offshore Cranes: TPK4560 - Specialisation Project, 18 Dec. 2020. Accessed 18 Dec. 2020.

[7] *Kraner Og Andre Løfteinnretninger - Terminologi - Krantyper, Tekniske Data, Generelle Uttrykk Og Krankomponenter*. NS-5516, standard.no, 1 Jan. 1987, www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=134779. Accessed 15 Oct. 2020.

[8] Vignesh Balasubramaniyan. “Understanding Offshore Lifting Operations and Engineering Analysis.” *Marine Insight*, 31 July 2020, www.marineinsight.com/offshore/offshore-lifting-operations-and-engineering-analysis/. Accessed 10 Nov. 2020.

[9] Rollins, Leo. “Robust Control Theory.” *Users.Ece.Cmu.Edu*, 1999, users.ece.cmu.edu/~koopman/des_s99/control_theory/. Accessed 9 Dec. 2020.

[10] Sciavicco, Lorenzo, and Bruno Siciliano. *Modelling and Control of Robot Manipulators*. 1996. 2nd ed., London ; New York, Springer, 2000.

[11] “10ft Storage Container.” *Container Technology, Inc*, containertech.com/container-sales/10ft-storage-container/. Accessed 14 Dec. 2020.

# 6. Appendices

List of external files:

- init_Crane.m
  - *Initialisation values for the crane simulation*

- calculations.m
  - *Calculations of rotational matrices*

- createRobotModel.m
  - *Script for generating a Simulink/Simscape robot model*

- arduinoTest.m
  - *Script for testing MPU6050 chip with both Raspberry Pi and Arduino*

- craneSim.urdf
  - *Universal Robot Description Format file describing the manipulators*

- Crane.slx
  - *Simulink simulation model*

- B_K_crane v5.f3d
  - *Autodesk Fusion 360 3D Design file of Boom an Knuckle Crane used for visualisation*