NTNU

Norwegian University of
Science and Technology

# Three-Axis Motion Compensated Crane Head Control

**Vegard Wie Henriksen**

**Audun Gerhardsen Røine**

Master of Science in Cybernetics and Robotics

|  |  |
|---|---|
| Submission date: | June 2016 |
| Supervisor: | Tor Arne Johansen, ITK |
| Co-supervisor: | Espen Skjong, ITK |

Norwegian University of Science and Technology
Department of Engineering Cybernetics

| | |
|---|---|
| **Title:** | Three-Axis Motion Compensated Crane Head Control |
| **Student:** | Vegard Wie Henriksen |
| **Student:** | Audun Gerhardsen Røine |

**Problem description:**

For ships at sea, the environmental forces of wind, waves and current can cause challenges for offshore crane operations. Historically, the main method of combating this is Active- or Passive Heave Compensation, which works to counteract the lateral movements of the ship, heave. The rocking movements of the ship, roll and pitch are harder to counteract, but new technology seeks to remedy this.

A new type of crane, using a Delta parallel robot as its crane head is being developed, and the overall goal of this project is to achieve motion compensation based on feedback from an Inertial Measurement Unit (IMU).

The students are to describe the mathematical model of the parallel robot crane head, and to derive control algorithms needed for motion compensation. To validate and to ensure that motion compensation is achieved, computer simulations should be done, as well as the construction of a prototype of the crane head, tested with a real life IMU.

| | |
|---|---|
| **Responsible professor:** | Professor Tor-Arne Johansen, ITK |
| **Supervisor:** | PhD Candidate Espen Skjong, ITK |

# Abstract

Offshore operations can be harsh and demanding and set personnel and equipment at risk. Ships will be exposed to the environmental forces of wind, waves and current, which will influence offshore crane operations considerably. This thesis addresses the use of a crane head, a Three Axis Compensator (TAC), constructed as a Delta parallel robot, to compensate for the motions of the ship in three axes. This type of robot has a rigid and accurate structure, and because of its highly nonlinear nature, advanced control algorithms must be derived. This thesis includes both forward and inverse kinematics for the robot, as well as velocity kinematics and workspace analysis. The kinematics of a full crane system, with the TAC as its head, has also been modelled. To evaluate the control algorithms and to prove the concept of using a Delta parallel robot for motion compensation, two products were made, a simulator and a prototype of the TAC.

A simulator which includes a model of a supply ship, a full scale crane with the TAC as its head has been created. The disturbances perturbed on the system from the elements are translated and rotated to the crane head frame of reference for use in the compensation procedure. PID-regulators are used to control the crane head, and simulations are conducted to verify that the crane head is able to compensate for the motions created by waves, tested at different sea states.

In addition a small-scale prototype of the TAC has been built. This robot uses two-phase stepper motors to control the position of the Tool Center Point (TCP) which is where the cargo will be suspended. The prototype's body is of 3D-printed plastic parts (ABS), magnets, steel balls and carbon tubes, and an Arduino microcontroller board is used to control the steppers.

The sensor used to track the movements of the TAC is an Inertial Measurement Unit (IMU), which is a unit used to measure linear acceleration and rotational velocity. These signals were thoroughly filtered to avoid measurement noise on the measurements required in the control algorithms. Because of the structure's highly non-linear nature, some of the calculations in the control algorithm are strenuous affairs and are conducted in MATLAB on a computer connected to the Arduino, due to high computational costs.

The subject matter described in this thesis, merging the Delta robot with an IMU into the TAC is a novel application, and as such has resulted in a conference paper at the 10th IFAC Conference on Control Application Marine Systems (CAMS) to be held in Trondheim, Norway September 13-16, 2016. The paper is included in Appendix C.

# Sammendrag

Marine operasjoner kan være krevende og sette både personell og utstyr i fare. Skip blir utsatt for vind, bølger og havstrømmer, noe som påvirker kranoperasjoner til havs i stor grad. Denne oppgaven tar for seg bruk av et kranhode, kunstruert som en Delta parallellrobot, til å kompensere for skipets bevegelser i alle tre akser. Denne typen roboters oppbygning gir den en kraftig og presis struktur, og siden den er høyst ulineær av natur må det avanserte kontrollalgoritmer til for å styre den. Både fremover- og inverskinematikken til roboten vil bli presentert i denne avhandlingen, sammen med hastighetskinematikken og en analyse av robotens arbeidsområde. I tillegg har en full skala kran som skal bruke robotten som kranhodet modellert. For å evaluere kontrollalgoritmene, og bevise konseptet av å bruke en Delta parallellrobot til bevegelseskompensering, ble to ting produsert, en simulator og en prototype av roboten.

En simulator som inkluderer en modell av et forsynigsskip, en full skala kran, og med roboten som kranhode har blitt produsert. Forstyrrelsene som påvirker systemet er translatert og rotert fra skipet til kranhodet for å brukes i kompenseringsprosedyren. En PID-kontroller er brukt for å kontrollere kranhodet, og simuleringene er gjort for å verifisere at kranhodet er i stand til å kompensere for bevegelser som er resultat av eksterne påvirkninger, og dette er testet ved forskjellige havforhold.

I tillegg ble en nedskalert prototype av kranhodet bygget. Roboten bruker to-fasede steppermotorer for å kontrollere posisjonen i tuppen av krana, som er punktet hvor en last er oppankret. Skjelettet til prototypen ble laget av 3D-printede plastikkdeler (ABS), magneter, stålkuler og karbonrør, og et Arduino mikrokontrollerbrett ble brukt for å kontrollere motorene.

Sensoren for å måle bevegelsene til kranhodet er en IMU, en enhet som måler linære akselerasjoner og rotasjonshastigheter. Disse målingene ble nøye filtrert for å unngå støy på nyttesignalet som ble benyttet i kontrollalgoritmene. Siden roboten er ulineær, og noen av utregningene som må gjøres i forbindelse med kontrollalgoritmene er tunge regneoperasjoner, ble mesteparten av utregningene gjort i MATLAB på en PC som var koblet til Arduinoen.

Fagstoffet som er beskrevet i denne oppgaven, sammenkobling og integrasjon av en Deltarobot med en IMU, er noe som ikke har blitt gjort før. Dette har resultert i en artikkel som ble skrevet til 10th IFAC Conference on Control Application Marine Systems (CAMS) som blir holdt i Trondheim fra 13. til 16. september 2016. Artikkelen kan bli funnet i vedlegg C.

# Preface

This thesis is submitted in partial fulfillment of the requirements for the degree MSc. at the Norwegian University of Science and Technology. The thesis is written by two collaborating students, Vegard Wie Henriksen and Audun Gerhardsen Røine, who have with the help of Department of Engineering Cybernetics (ITK), completed all the work described within. The thesis builds on a fall project both students undertook seperately in the fall of 2015.

This project has been done in cooperation with Ulstein Power & Control AS (UPC), and is a chance for the authors to help the development of a creative new type of crane that Ulstein has come up with, and to explore the possibilites of this type of contraption. This thesis seeks to enlighten the reader on the accomplishments and discoveries made during the course of the last six months, which has resulted in mathematical models, simulations, small-scale microcontroller governed robots, as well as a conference paper accepted to the 10th IFAC Conference on Control Application Marine Systems (CAMS) to be held in Trondheim, Norway during September 13-16, 2016

We would like to thank our advisors and paper co-authors, Prof. Tor-Arne Johansen and PhD. cand. Espen Skjong at ITK for all the guidance and motivation they have given us. Also we thank the students we have shared offices with for the last year for all the good times. Finally we would like to thank our families and partners for all support and encouragement necessary to complete our goal.

# Contents

Appendices

# List of Figures

# List of Tables

# List of Algorithms

# Abbreviations

| | |
|---|---|
| **ABS** | Acrylonitrile Butadiene Styrene |
| **AHC** | Active Heave Compensation |
| **AHRS** | Attitude and Heading Reference System |
| **CO** | Coordinate Origin |
| **DOF** | Degrees Of Freedom |
| **DP** | Dynamic Positioning |
| **EMC** | ElectroMagnetic Compatibility |
| **FFT** | Fast Fourier Transform |
| **FPK** | Forward Position Kinematics |
| **GNSS** | Global Navigation Satellite System |
| **HIL** | Hardware-In-the-Loop |
| **HPF** | High Pass Filter |
| **IMU** | Intertial Measurement Unit |
| **I2C** | Inter-Integrated Circuit |
| **INS** | Inertial Navigation System |
| **IPK** | Inverse Position Kinematics |
| **LARS** | Launching And Retrieving Systems |
| **LPF** | Low Pass Filter |
| **MEMS** | Micro Electric Mechanical Systems |
| **MPC** | Model Predictive Control |
| **NED** | North East Down frame |
| **SPI** | Serial Peripheral Interface |
| **TAC** | Three-Axis Compensator |
| **TCP** | Tool Centre Point |
| **UART** | Universal Asynchronous Receiver/Transmitter |

# Mathematical Notation and Definitions

- *Matrix* - $\mathbf{A}$ given on the form

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & ... & a_{1,n} \\ a_{2,1} & a_{2,2} & ... & ... \\ ... & ... & ... & ... \\ a_{m,1} & ... & ... & a_{m,n} \end{bmatrix}$$

- *Vector* - $\mathbf{v}$ given on the form $\mathbf{v} = [x, y, z]^T$
- *Coordinate* - $\mathbf{p}$ given on the form $\mathbf{p} = (x, y, z)^T$
- *Scalar* - $l$
- *Reference frame* - $\{n\}$
- $\sin(x)$ - $s_x$
- $\cos(x)$ - $c_x$
- *Rotation matrices* are 3x3-matrices consisting of the rotation vectors around the $x$-, $y$- and $z$-vector on the form

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}_{3x1} & \mathbf{y}_{3x1} & \mathbf{z}_{3x1} \end{bmatrix}$$

  Rotation matrices for the euler angles roll, pitch and yaw ($\phi$, $\theta$ and $\psi$) are on the form -

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}, \quad \mathbf{R}_{y,\theta} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \text{ and } \mathbf{R}_{z,\psi} = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

- *Skew symmetric matrices* -

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix},$$

  where $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$.

- *Derivative of a rotation matrix* -

$$\dot{\mathbf{R}}(t) = \mathbf{S}(\boldsymbol{\omega})\mathbf{R}(t),$$

- *Transformation matrix* -

$$T = \begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{o}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}$$

  where $\mathbf{R}$ represents the rotational change in attitude, and $\mathbf{o}$ represents the translational change in position.

# Chapter 1
# Introduction

## 1.1 Background

Offshore crane operations in harsh environments are challenging and put crew and equipment at risk. Heave compensated crane systems in marine vessels have been extensively used to cultivate easier and safer offshore operations. Examples of such operations are surface crane operations for installing equipment at the seafloor, launching and retrieving systems (LARS) and delivering supplies from vessels to platforms. Motion compensation in such systems are generally limited to one axis, i.e. the vertical heave motion, [7, 9, 10, 12].

This thesis considers the development of a crane head designed for motion compensation in all three axes. This Three Axis Compensator (TAC) is a Delta type parallel robot [3]. A parallel robot consists of two or more closed kinematic chains linking the base to the end effector, whereas its counterpart, the serial robot consists of just one kinematic chain [18]. The advantages of a parallel structure is its high rigidity and accuracy [11], making it very attractive for crane operations, whereas the disadvantages are narrower workspace and more difficult control than a serial robot [11]. The Delta robot consists of three kinematic chains connected on either end at a top- and bottom plate, and these plates stay in parallel with each other [4]. It is most commonly used for precise and stationary actions such as item picking or 3D printing [21], but in this thesis it will be seen that it can also be used for motion compensation of marine crane operations, which is a novel application.

The TAC is based upon the design made by the Colibri three axis crane head system, seen in Figure 1.1, designed by the Dutch sub-division Ulstein Idea of Ulstein in collaboration with National Oilwell Varco. As MacGregor recently revealed they have built the world's most stable crane [20], a crane which uses a large multi-purpose platform which holds the weight of the crane and the cargo to compensate for the movement, Ulstein is taking a different approach as the Colibri only carries the weight of the cargo.

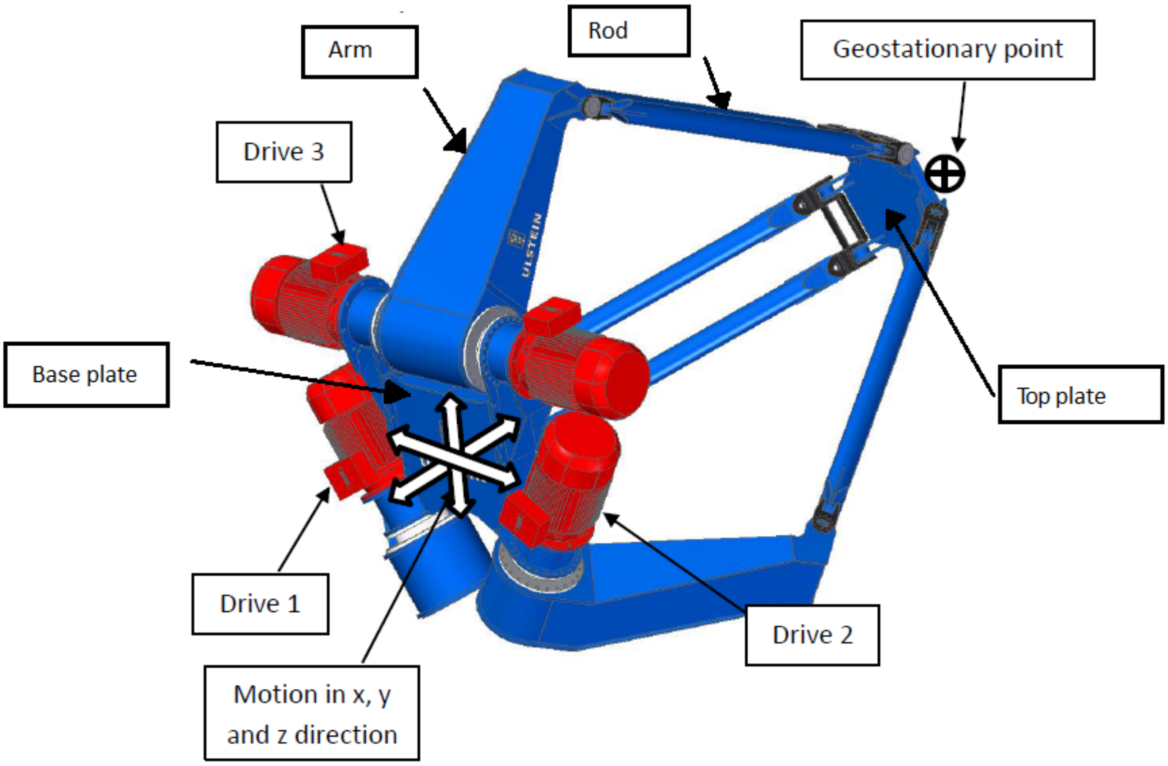


**Figure 1.1:** Schematic illustration of the Colibri Crane head

The Colibri is controlled by changing the angle of the three lower arms to the base plate, called joint angles. DC motors placed at the base of the arms are used to control the joint angles. As seen in Figure 1.1, the top drive contains two engines instead of one. This is to compensate for gravitational forces. The input signal to the DC motor will be the desired angle. The Colibri could be used in collaboration with other control methods to increase accuracy and workspace for three-axis compensation, however it can also be used as a standalone device to actively compensate for the ship's movements affecting the crane operations.

## 1.2 Project Description

The overall goal of this project is to use the TAC for motion compensation. As the motion of the ocean can be hard to predict, it must be measured, which is to be done with an IMU. To achieve motion compensation, the geometry and dynamics of the TAC need to be derived, and control algorithms for the crane system for precise and timely compensation need to be designed. Closing the feedback loop using an IMU requires the filtering of the measurements to suppress enough noise for the signal to be useful.

This goal will be achieved in two ways: First, a model of a dynamic system including waves, a ship and a crane with the TAC as its head will be created. Simulations will be conducted to discover what type of ocean conditions the crane can be used to compensate for the motion of cargo hanging from it. Secondly, a small-scale prototype of the TAC shall be created. This prototype shall act as a proof of concept of the compensation capabilities of the TAC, and utilized to achieve realistic experimental data.

## 1.3 Report Structure

This section will present the structure of the thesis.

- **Chapter 2** presents theory relevant to the project described in this thesis. Theory within robotics, navigation, control systems, and signal processing will be presented.
- **Chapter 3** presents theory connected the full crane system including the TAC used in this project. The geometric and kinetic relationships of the crane , which will be useful tools to control the system, will be derived.
- **Chapter 4** details the simulator developed in MATLAB/Simulink to test the model derived out in chapter 3. The control algorithm for the setup with a three-jointed elbow manipulator as the crane's body with the TAC as its end effector will be derived, and the results from the simulations will be presented.
- **Chapter 5** describes the prototype developed to prove the concept of motion compensation using a Delta robot with an IMU as sensor. Components, circuit schematics, wire list, control algorithms and noise filtering will be presented
- **Chapter 6** shows results from the prototype system. This includes plots of the signal processing, as well as plots that show the results from the motion compensation tests of the prototype.
- **Chapter 7** discusses the different solutions chosen to complete this project, possible improvements that could be made, and the results from testing.
- **Chapter 8** acts to tie together the threads picked up in the thesis, and to highlight the conclusions drawn.

Chapter 2

# Background Theory

Well known topics like robotics, navigation and signal processing are keys to reach the thesis objective. This chapter will describe much of the background theory this thesis is based on.

## 2.1 Robotics

Describing robotic systems with several joints can be a strenuous and nonlinear affair. The tasks detailed in this report requires the use of many different robotic methods, including forward kinematics, inverse kinematics and velocity kinematics which will be described in this section.

### 2.1.1 Robotic Structure

A robotic structure is composed of links connected by joints to form a kinematic chain [18]. The typical type of joints are either revolute or prismatic. A prismatic joint allows a linear relative motion between two links, whereas a revolute joint acts like a hinge and allows relative rotation between two links. For this project, only revolute joints are used. Figure 2.1 shows an example of a kinematic chain consisting of three links and two joints. The first link is fastened to the ground while the other two can be revolved with the joint angles $\theta_1$ and $\theta_2$. Each link has its own base coordinate frame, $\mathbf{o}_i = (x_i, y_i)$, $\forall i \in \{0, 1, 2\}$, and the transformations between these frames are the focus of both forward and inverse kinematics. The Tool Centre Point (TCP) is the end of the kinematic chain, in Figure 2.1 this is $\mathbf{o}_2 = (x_2, y_2)$, and is where the end-effector of the robot would be. The joint angles of the robot can be gathered in the vector $\boldsymbol{q}$, which for Figure 2.1 can be stated as $\boldsymbol{q} = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^T$.

### 2.1.2 Forward Kinematics

Forward kinematics is the method of describing the position of the TCP by the joints. A tool for simplifying this process significantly is the Denavit-Hartenberg (DH) convention [18]. The DH convention states that the homogeneous transformation for any link $i$ is represented as the product of four basic transformations, a rotation $\theta_i$ about the $z_{i-1}$-axis, a translation $d_i$ along the $z_{i-1}$-axis, a translation $a_i$ along the $x_{i-1}$-axis, and finally a rotation $\alpha_i$ about the $z_{i-1}$-axis. These parameters are handily placed inside a DH-table, as in Table 2.1, for $i = 1, 2, ..., N$ joints. A transformation matrix is a 4×4 matrix consisting of either a rotation, a translation or both.

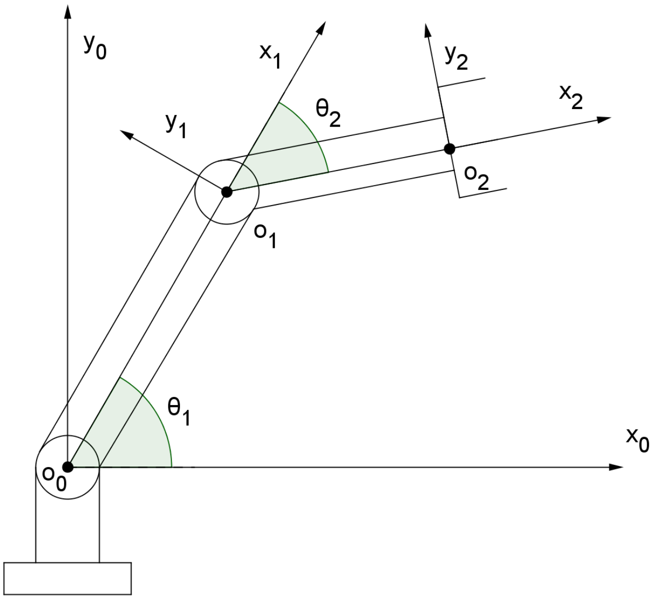


**Figure 2.1:** Kinematic chain with three links

**Table 2.1:** DH table example

| **Link** | $\boldsymbol{a_i}$ | $\boldsymbol{\alpha_i}$ | $\boldsymbol{d_i}$ | $\boldsymbol{\theta_i}$ |
|---|---|---|---|---|
| 1 | $a_1$ | $\alpha_1$ | $d_1$ | $\theta_1$ |
| 2 | $a_2$ | $\alpha_2$ | $d_2$ | $\theta_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| N | $a_N$ | $\alpha_N$ | $d_N$ | $\theta_N$ |

In (2.1) it is shown how the DH matrix $A_i$ is derived from the four basic transformation matrices

$$\begin{aligned}\boldsymbol{A_i} &= \boldsymbol{Rot}_{z,\theta_i}\boldsymbol{Trans}_{z,d_i}\boldsymbol{Trans}_{x,a_i}\boldsymbol{Rot}_{z,\alpha}\\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0\\ s_{\theta_i} & c_{\theta_i} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & d_i\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & a_i\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & a_i\\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0\\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ic_{\theta_i}\\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_is_{\theta_i}\\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i\\ 0 & 0 & 0 & 1\end{bmatrix}\end{aligned} \tag{2.1}$$

The notation for $\sin(\cdot)$ and $\cos(\cdot)$ in (2.1) is $s_x \equiv \sin x$ and $c_x \equiv \cos x$. The transformation matrix between frames can be found as

$$\mathbf{T}_n^0 = \mathbf{A}_0 \cdots \mathbf{A}_n \tag{2.2}$$

where $n$ is the number of joints. The transformation matrix can be further broken down as

$$\mathbf{T}_n^0 = \begin{bmatrix} \boldsymbol{R}_n^0 & \boldsymbol{o}_n^0 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_n^0 & \boldsymbol{y}_n^0 & \boldsymbol{z}_n^0 & \boldsymbol{o}_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where $\boldsymbol{R}_n^0$ is a rotation matrix consisting of the three vectors $\boldsymbol{x}_n^0$, $\boldsymbol{y}_n^0$ and $\boldsymbol{z}_n^0$ , and $\boldsymbol{o}_n^0$ is the translation vector.

### 2.1.3 Inverse Kinematics

The inverse kinematics is a method of finding the joint variables, $\boldsymbol{\theta}$, with the knowledge of the TCP position. The inverse kinematics is in general more difficult than the forward kinematics [18], and the solutions can be very different for seemingly similar constructions. In this thesis the inverse kinematics of both an elbow manipulator and the TAC be will be detailed in Chapter 3.

### 2.1.4 Velocity Kinematics

The velocity kinematics of the robot is the relationship between joint speed, $\dot{\boldsymbol{q}}$, and the velocity of the end effector, $\boldsymbol{\xi}$, described with by

$$\boldsymbol{\xi} = \boldsymbol{J}\dot{\boldsymbol{q}}, \tag{2.4}$$

where $\boldsymbol{J}$ is the manipulator Jacobian matrix [18]. The velocity vector can be split into the linear velocity, $\boldsymbol{v}$, and the angular velocity, $\boldsymbol{\omega}$. Correspondingly, the Jacobian splits into the linear velocity Jacobian, $\boldsymbol{J}_v$, and the angular velocity Jacobian, $\boldsymbol{J}_\omega$

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}, \quad \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_v \\ \boldsymbol{J}_\omega \end{bmatrix}, \tag{2.5}$$

where both $\boldsymbol{J}_v$ and $\boldsymbol{J}_\omega$ have one column for each joint, i.e.

$$\begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{v1} ... \mathbf{J}_{vn} \\ \mathbf{J}_{\omega 1} ... \mathbf{J}_{\omega n} \end{bmatrix}. \tag{2.6}$$

For revolute joints, the $i^{th}$ column of the Jacobian can be stated as

$$\begin{bmatrix} \mathbf{J}_{v_i} \\ \mathbf{J}_{\omega_i} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}), \\ \mathbf{z}_{i-1}. \end{bmatrix}. \tag{2.7}$$

The Jacobian can be a useful tool in robotics. For parallel robots, such as the TAC, finding the Jacobian is not as straightforward, as will be seen in Chapter 3.

### 2.1.5 Workspace

The workspace of a robot is the total volume reachable by the TCP [18]. It is an important aspect of robotics to have a well defined workspace, since trying to reach something outside of it could distort the robot into impossible configurations that may have dire and expensive repercussions. Deriving the workspace is generally more difficult the more joints there are in the robot, and the more complex the robot is.

## 2.2 Navigation and Control

Doing motion compensation at sea requires good measurements of attitude and linear displacement. This section presents the methods of describing the motion of a ship at sea, and the decomposing of movements into Degrees of Freedom (DOF). Also the main measurement device used in this project, the Inertial Measurement Unit (IMU), will be described.

### 2.2.1 Reference Frames and Degrees of Freedom

When working with kinematics and kinetics it is useful to define some reference frames. The North-East-Down (NED) frame is defined as the tangent plane on the Earth's surface moving with the vessel. It is denoted $\{n\} = (x_n, y_n, z_n)$ and has origin $o_n$ defined relative to the Earth's reference ellipsoid. The x-axis is pointing to true North, y axis is pointing East and the z-axis is pointing down towards the core of the Earth.
The body frame is defined as a moving coordinate frame fixed to the vessel. This frame is denoted $\{b\} = (x_b, y_b, z_b)$ with origin $o_b$ usually defined as the center of gravity. $o_b$ is often called $CO$. In the body frame $x_b$ is pointing forward, $y_b$ is pointing starboard and $z_b$ is pointing downwards.

Marine vessels are affected by forces and moments in six DOF [8], which are further described in Table 2.2. This means that the ship's movements can be decomposed into 6 movements where 3 of them are translations and the other 3 are rotations. These variables will be expressed in the vector form

$$\begin{aligned} \boldsymbol{\eta} &= \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T, \\ \boldsymbol{\nu} &= \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T. \end{aligned} \tag{2.8}$$

These DOF's can be recognized as the difference between frame $\{n\}$ and $\{b\}$. The six DOF's are illustrated in Figure 2.2.

| DOF | | Linear & angular velocities | Positions & Euler angles |
|---|---|---|---|
| 1 | translations $x$ dir (surge) | $u$ | $x$ |
| 2 | translations $y$ dir (sway) | $v$ | $y$ |
| 3 | translations $z$ dir (heave) | $w$ | $z$ |
| 4 | rotation $x$ axis (roll) | $p$ | $\phi$ |
| 5 | rotation $y$ axis (pitch) | $q$ | $\theta$ |
| 6 | rotation $z$ axis (yaw) | $r$ | $\psi$ |

**Table 2.2:** Notation for marine vessels [17]

### 2.2.2 Inertial Measurement Unit

An IMU is a unit that provides three axis acceleration and turning rate detection [13] and is widely used for navigation and control. The IMU normally consists of accelerometers, gyroscopes and magnetometers, among other sensors, and can be used to estimate position and attitude, and the velocities of these [8].

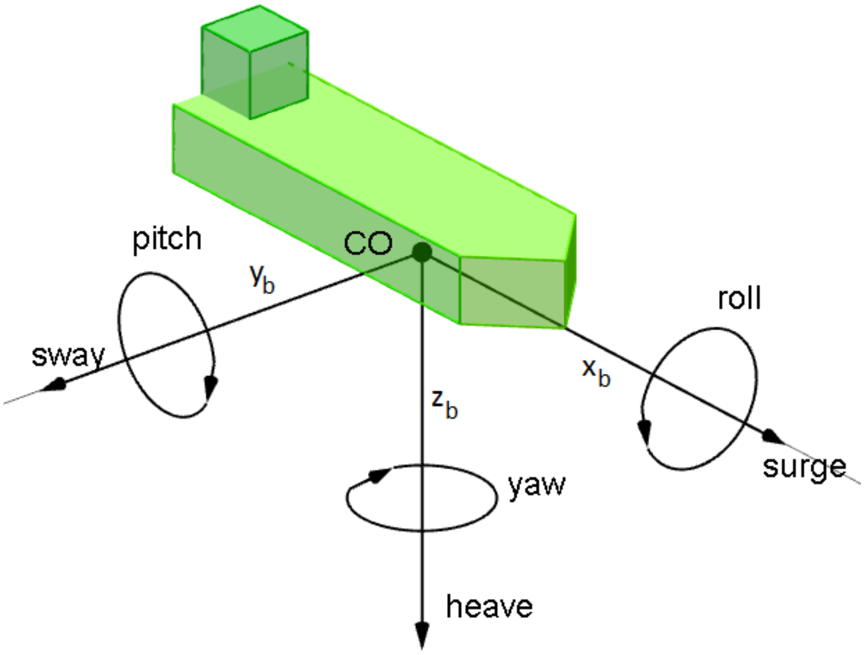


**Figure 2.2:** Demonstration of the six DOF's on a ship

Accelerometers are used to measure linear acceleration. There are multiple types of accelerometers, two of them are mechanical and vibrational accelerometers. Mechanical accelerometers can be as simple as a pendulum where the measurements are based on Newtons second law, $F = ma$. The mass $m$ of the instrument is known and the acting force $F$ is measured, thus $a$ can easily be calculated. Vibrational accelerometers on the other hand is for example a tense, vibrating string. When a force is acting on this string the tension change together with the vibration frequency. Since acceleration is proportional to the difference in frequency, the acceleration can be calculated. Note that both the mechanical and vibrational accelerometers can be other devices than described here. Assuming the IMU is placed close to the coordinate origin (CO) the linear accelerations are given by

$$\boldsymbol{a}^b_{imu} = \boldsymbol{R}^b_n(\boldsymbol{\Theta})(\dot{\boldsymbol{v}}^n_{m/n} + \boldsymbol{g}^n) + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \tag{2.9}$$

where $\boldsymbol{\Theta} = [\phi \quad \theta \quad \psi]^T$ is the Euler angles and $\boldsymbol{R}^b_n(\Theta)$ denotes the rotation between {n}-frame and {b}-frame. $\dot{\boldsymbol{v}}^n_{m/n}$ is the estimated linear acceleration vector and $\boldsymbol{g}^n = [0 \quad 0 \quad 9.80665]^T \quad m/s^2$ is the nominal average gravity vector. $\boldsymbol{b}^b_{acc}$ is the accelerometer bias and $\boldsymbol{w}^b_{acc}$ is measurement noise. The measured linear acceleration can be integrated one time to get an estimation of linear velocity and two times to get linear position. Sensor biases, misalignments and temperature differences can cause drift, which causes the estimates to be unreliable. One method for removing measurement noise from a signal is a low pass digital filter, whereas a high-pass filter can be used to remove a signal offset. Filtering will be discussed in more details in section 2.3.

Gyroscopes are used to measure angular velocity. The classic gyroscope consists of a spinning wheel mounted in gimbal rings, which lets the wheel turn freely in any direction. Due to the momentum of the spinning wheel, its attitude does not change when the surroundings change.

This effect can be used to estimate rotation. Other types of gyros include optical gyros such as ring laser gyros and fiber optic gyros, and also gyros based on micro-electromechanical systems (MEMS). The angular speed measurements are given by

$$\boldsymbol{\omega}^b_{imu} = \boldsymbol{\omega}^b_{m/n} + \mathbf{b}^b_{gyro} + \mathbf{w}^b_{gyro} \tag{2.10}$$

where $\boldsymbol{\omega}^b_{m/n}$ is the estimated angular velocity vector, $\mathbf{b}^b_{gyro}$ as the gyro bias and $\mathbf{w}^b_{gyro}$ is the measurement noise from the gyro.

A magnetometer is an instrument that measures magnetic fields, and in navigational purposes it is used to measure the Earth's magnetic field. This measurement is combined with the other IMU measurements to provide more accurate heading angles for IMUs. The measurements from the magnetometer are given by

$$\mathbf{m}^b_{imu} = \mathbf{R}^b_n(\boldsymbol{\Theta})\mathbf{m}^n + \mathbf{b}^b_{mag} + \boldsymbol{w}^b_{mag} \tag{2.11}$$

where $\mathbf{m}^n$ is the magnetometer measurement, $\mathbf{b}^b_{mag}$ is the magnetometer bias, $\mathbf{w}^b_{gyro}$ is the measurement noise from the magnetometer and $\mathbf{R}^b_n$ transforms the measurements from frame {n} to {b}. Furthermore, the calibrated magnetometers measurements are given by $[m_x \quad m_y \quad m_z]^T = \mathbf{m}^b_{imu} - \mathbf{b}^b_{mag}$. This can be used to find the magnetic heading angle $\psi_m = \arctan(m_y/m_x)$.

To obtain measurements of higher quality the IMU can be combined with other systems. Combining the IMU with the kinematic equations for the ship results in an Inertial Navigation System (INS). By including Global Navigation Satellite System (GNSS) measurements in the computations the estimation drift can be removed. The acceleration measurement quality depends on the IMU quality while the velocity and position measurement quality depends on the GNSS quality [8].

## 2.3 Signal Processing

Measured signals can be corrupted by measurement noise or a bias, and may need signal processing before they can be used for anything important. This section will present a way of filtering the signals and a method for discrete integration of signals.

### 2.3.1 Filters

Mainly filters are divided into Low Pass Filters (LPF), High Pass Filters (HPF), band stop filters, band pass filters and combination of these are widely used. LPFs are designed to let the signals of low frequencies through and suppressing the high frequencies. HPFs, on the other hand, let through the high frequency signals and suppress low frequencies. The frequency where the filter is going to suppress signals from or to, is called the cutoff frequency, $f_c$, and is defined as the frequency where the signal is reduced with 3dB due to the filter [16]. Filters are using both old and new measurements and old estimated values to estimate the current value. How many steps the filter looks back when estimating the current value decides the filter order. The filter transfer function for a first order Butterworth filter in the Z-domain is given by

$$H(z) = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}}, \tag{2.12}$$

where $a$ and $b$ are filter constants designed such that the filter has the wanted cutoff frequency. The equation is similar for all first order Butterworth filters, but $a$ and $b$ decides if it is a HPF or LPF. This filter can be realized digitally by

$$a_0 y_k = b_0 x_k + b_1 x_{k-1} - a_1 y_{k-1}. \tag{2.13}$$

This is, as mentioned before, a first order filter that is using measurements and estimates from one step back. A Butterworth filter of the n'th order can be realized digitally by

$$a_0 y_k = b_0 x_k + b_1 x_{k-1} + ... + b_n x_{k-n} - a_1 y_{k-1} - ... - a_n y_{k-n}. \tag{2.14}$$

Filters of higher order are slower than filter of lower orders, but has a steeper ramp from cutoff to stop band or pass band. Figure 2.3 compares the magnitude of Butterworth LPF of order 1 to 11. As can be seen of the plot the filter of order 11 has a much steeper cutoff than the filter of order 1. A steeper cutoff means that less of the signal with frequencies smaller than the cutoff frequency is considered in the filtered signal for high order filters. It can also be seen that part of the signal with frequencies higher than, but close to, the cutoff frequencies is suppressed by low order filters. This is for LPF and the reverse way for HPF.

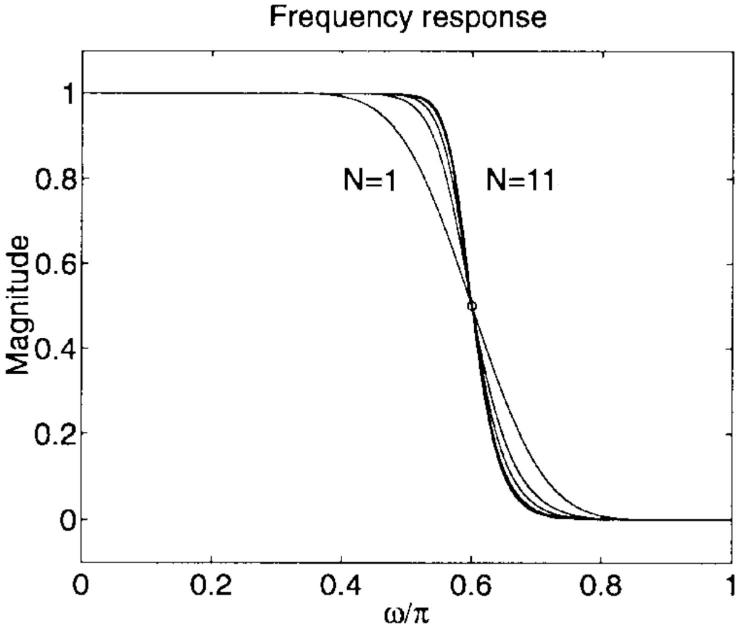


**Figure 2.3:** Comparison of steepness in cutoff region for Butterworth LPF of different order [15]

### 2.3.2 Filter design

This procedure used to decide which and what type of filters to be used can be summed up to

- If the measurement has a lot of noise compared to the desired signal, use a LPF
- If the measurement has a bias compared to the desired signal, use a HPF

- Analyze the magnitude spectrum of the measurement to find fitting cutoff frequencies for filters

The magnitude spectrum of a signal $x(t)$ is found by first doing a Fourier transform [16]

$$X(f) = FFT(x(t)), \tag{2.15}$$

then the normalized magnitude spectrum, $|X_N(f)|$, is found by

$$|X_N(f)| = \frac{|X(f)|}{N}, \tag{2.16}$$

where $N$ is the number of samples. This will give a two-sided spectrum with both negative and positive frequencies, but for this project only the positive frequencies will be looked at. This one-sided spectrum can be plotted, and the cutoff frequencies be decided by seeing what frequencies the desired signals operate in.

### Example of filter design

The desired signal is a sinus wave, $d(t) = \sin(5t)$. To simulate a measurement with noise the desired signal will be disturbed both by zero mean white noise $w(t)$ with a max amplitude of 1, and a linear drift of $0.5t$. The resulting measured signal $x(t)$ is then given by

$$x(t) = 0.5t \cdot d(t) + w(t). \tag{2.17}$$

Figure 2.4 illustrates the magnitude spectrum of the measured signal $x(t)$. By analyzing the magnitude spectrum it can be seen that the desired signal probably has a frequency around 1 Hz, while the frequencies over 3 Hz looks like random noise and close to 0 Hz it looks to be because of the drift. To get rid of the noise a first order Butterworth LPF with with $f_c = 3$Hz was chosen, whereas to get rid of the bias HPF with $f_c = 0.5$Hz was chosen.

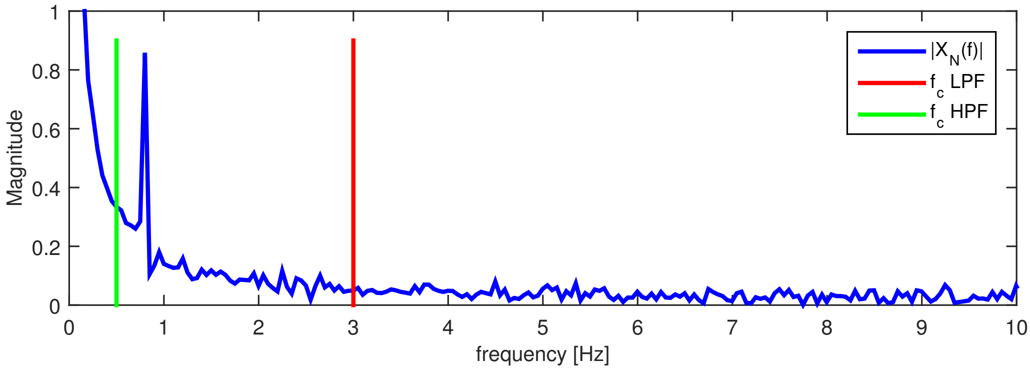


**Figure 2.4:** Magnitude for filter example

Figure 2.5 shows three plots. The first one shows the measured signal, compared to the desired signal. The second plot is showing the measurement compared to it after ran through the LPF. As can be seen, the low passed filtered signal is smoother than the original signal, but it still has an offset. The third plot is showing the signal when it is both low- and high pass filtered. The resulting filtered signal is both smooth and similar to the original signal and a good estimate to use in further calculations. The estimate has a phase delay compared to the desired signal, which is difficult to avoid when doing real time filtering. This phase delay will increase when the filter order increase. Alternatively the phase shift can be estimated and compensated for using a suitable estimator.

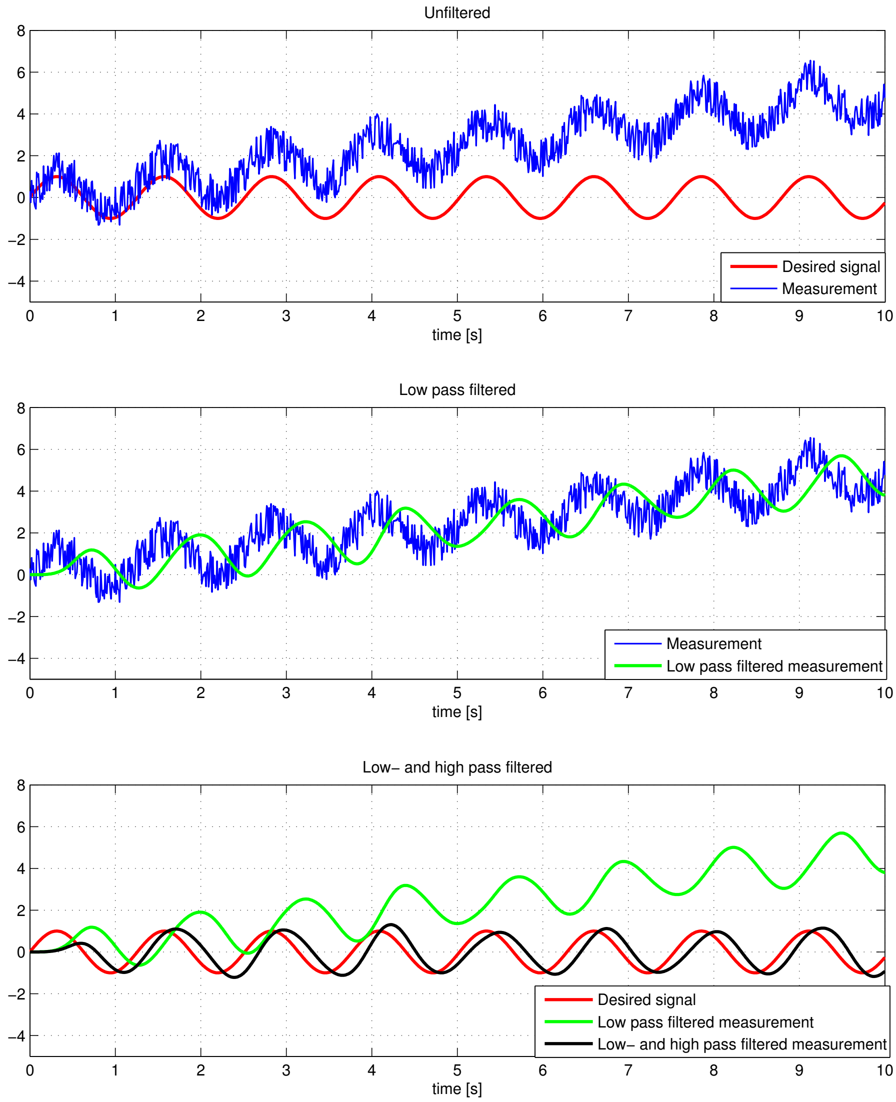


**Figure 2.5:** Filtering of test signal

### 2.3.3 Integration

Euler's Method is a simple but important numerical integration scheme [6]. Euler's method is given by

$$y_{n+1} = y_n + hf(t_n, y_n), \tag{2.18}$$

where $f(t_n, y_n) = \dot{y}$ and $h$ is step length. Euler's method can be useful if $y$ is unknown, while $\dot{y}$ is known, for example if the acceleration $a$ is known, but velocity $v$ is required, since $a = \dot{v}$.

# Chapter 3 Crane Description

The objective of the ship crane described in this thesis is motion compensation of suspended loads. For this to be achieved, knowledge about how the crane moves is paramount, and in this chapter a mathematical analysis of the crane system will be derived.

## 3.1 Three Axis Compensator

In this section the TAC's geometry will be detailed, the highly nonlinear kinematics of the structure derived, and a method of describing the workspace of the TAC will be presented.

### 3.1.1 TAC Geometry

Figure 3.1 shows a geometrical representation of the TAC with the different parameters further explained in Table 3.1. As this is a the head of a crane, the TCP is where the cargo is suspended, and its position is denoted $\mathbf{p}_c$. The position of the TCP is controlled by changing the angles between the base plate and the arms with the use of DC-motors. Figure 3.2 details the different reference frames used to describe the TAC's geometry. The main frame of orientation, denoted $\{t\}$ with coordinates $(x_t, y_t, z_t)$, is shown in Figure 3.2a, with the $x$-axis pointing out of the paper plane, $z$-axis up and the $y$-axis to the right. In Figure 3.2b the individual frame for any joint is shown, denoted $\{t_i\}$= $(x_{t_i}, y_{t_i}, z_{t_i})$ for $i \in \{1, 2, 3\}$, where the $y$-axis points from the knee into the center. Both {t}- and $\{t_i\}$-frames have the same origin, $\mathbf{o}_t$, thus transforming between these

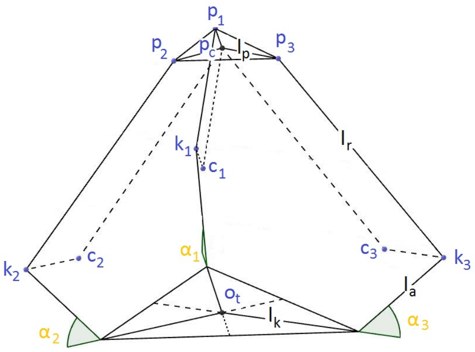
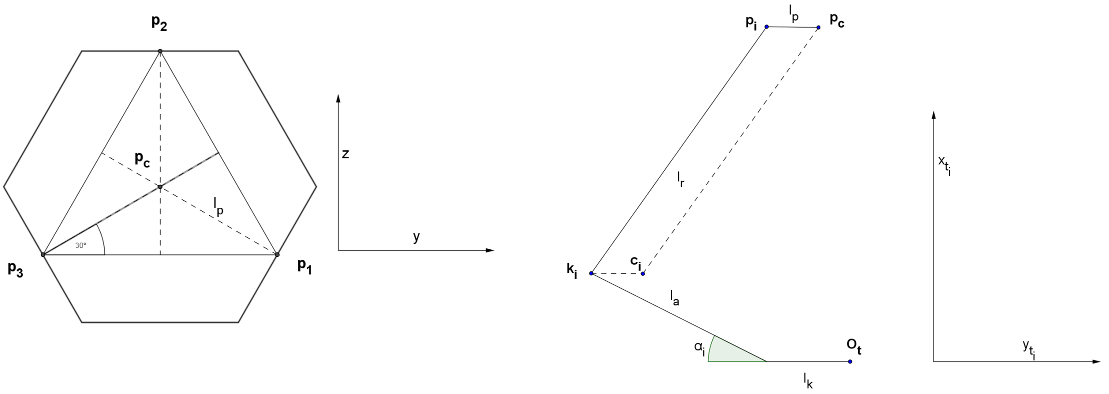


**Figure 3.1:** TAC Geometry

**Table 3.1:** TAC parameters

| **Notation** | **Unit** | **Description** |
|---|---|---|
| $l_k$ | m | Length from center of the bottom plate to the arm |
| $l_a$ | m | Length of the arm |
| $l_r$ | m | Length of the rod |
| $l_p$ | m | Length from the center of the top plate to the rod |
| $\alpha_i$ | rad | Angle between bottom plate and arm $i$ |
| $u_i$ | rad | input to DC-motors |
| $\mathbf{k}_i$ | - | Knee-point between arm and rod $i$ |
| $\mathbf{c}_i$ | - | Indented Knee-point $i$ |
| $\mathbf{p}_i$ | - | Point connecting rod and top plate |
| $\mathbf{p}_c$ | - | Center-point of the top plate, position of the TCP |
| $i$ | - | Jointed-arm number, $i \in \{1, 2, 3\}$. |



**(a)** TAC top plate seen from above, frame $\{t\}$ **(b)** TAC arm $i$ seen from one of the sides, frame $\{t_i\}$

**Figure 3.2:** The two main main frames of orientation for the TAC

frames is done by rotations, with the rotation matrices

$$\mathbf{R}^t_{t_1} = \mathbf{R}_{x,\frac{5\pi}{6}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}, \quad \mathbf{R}^t_{t_2} = \mathbf{R}_{x,\frac{3\pi}{2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix},$$
$$\text{and } \mathbf{R}^t_{t_3} = \mathbf{R}_{x,\frac{5\pi}{6}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{3.1}$$

The position of the TCP in the different frames is denoted as

$$\mathbf{p}_c^t = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T, \quad \text{and } \mathbf{p}_c^{t_i} = (\mathbf{R}_{t_i}^t)^T \mathbf{p}_c^t = [x_{ci} \quad y_{ci} \quad z_{ci}]^T, \quad \forall i \in \{1,2,3\}. \tag{3.2}$$

Each of the three kinematic chains consists of a arm and rod, connected by a knee joint. The position of the knee, $\mathbf{k}_i$, can be derived, when knowing the corresponding angle $\alpha_i$, as

$$\mathbf{k}_i^t = \mathbf{R}_{t_i}^t \begin{bmatrix} l_a \cos\alpha_i & -l_k - l_a \sin\alpha_i & 0 \end{bmatrix}^T \quad \forall i \in \{1,2,3\}, \tag{3.3}$$

whereas the position of the top plate cannot be found without knowing all three angles. The length of the rod, $l_r$, is constant, a fact that can be exploited to derive the relation between all angles and the TCP. By placing an indented knee-point, $\mathbf{c}_i$, a distance of $l_p$ in the $y_{t_i}$-direction in the $\{t_i\}$-frame yields a point which will be a constant distance of $l_r$ from $\mathbf{p}_c$. $\mathbf{c}_i$ can be described as

$$\mathbf{c}_i^t = \mathbf{R}_{t_i}^t \,[l_a \sin\alpha_i \quad a - l_a \cos\alpha_i \quad 0]^T, \quad \forall i \in \{1,2,3\}, \tag{3.4}$$

where $a = l_p - l_k$. This yields the three individual vectors in the {t}-frame

$$\mathbf{c}_1^t = \begin{bmatrix} l_a \sin\alpha_1 & -\frac{\sqrt{3}}{2}(a - l_a \cos\alpha_1) & \frac{1}{2}(a - l_a \cos\alpha_1) \end{bmatrix}^T, \tag{3.5}$$

$$\mathbf{c}_2^t = \begin{bmatrix} l_a \sin\alpha_2 & 0 & -(a - l_a \cos\alpha_2) \end{bmatrix}^T \text{ and} \tag{3.6}$$

$$\mathbf{c}_3^t = \begin{bmatrix} l_a \sin\alpha_3 & \frac{\sqrt{3}}{2}(a - l_a \cos\alpha_3) & \frac{1}{2}(a - l_a \cos\alpha_3) \end{bmatrix}^T. \tag{3.7}$$

The vector $\mathbf{s}_i$ is defined as the vector from $\mathbf{c}_i$ to $\mathbf{p}_c$ as

$$\mathbf{s}_i^t \equiv \mathbf{p}_c^t - \mathbf{c}_i^t \quad \forall i \in \{1,2,3\}. \tag{3.8}$$

$\boldsymbol{p}_c^t$ can be seen as the crossing point of three spheres with radius $l_r$, and center in each indented knee point $\boldsymbol{c}_i^t$, as presented in Figure 3.3. With this information, the vector-loop closure equation can be found

$$\|\mathbf{s}_i^t\|_2^2 = l_r^2 \quad \forall i \in \{1,2,3\}, \tag{3.9}$$

where the norm has been squared to avoid the root in the equation. Eq. (3.9) is a useful tool for describing the system dynamics [1, 4, 21]

### 3.1.2 Inverse Position Kinematics

The Inverse Position Kinematics (IPK) solution of the system is a way of finding the joint angles, $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T$, given the Cartesian coordinates of the TCP, $\mathbf{p}_c^t$ [21]. This is done in the $\{t_i\}$-frame, where the vector $\mathbf{c}_i^{t_i}$ is needed, denoted as

$$\mathbf{c}_i^{t_i} = [x_i \quad y_i \quad z_i]^T. \tag{3.10}$$

Eq. (3.9) can be expanded to

$$\|\mathbf{p}_c^{t_i}\|_2^2 + \|\mathbf{c}_i^{t_i}\|_2^2 - l_r^2 - 2(x_{ci}x_i + y_{ci}y_i + z_{ci}z_i) = 0, \tag{3.11}$$

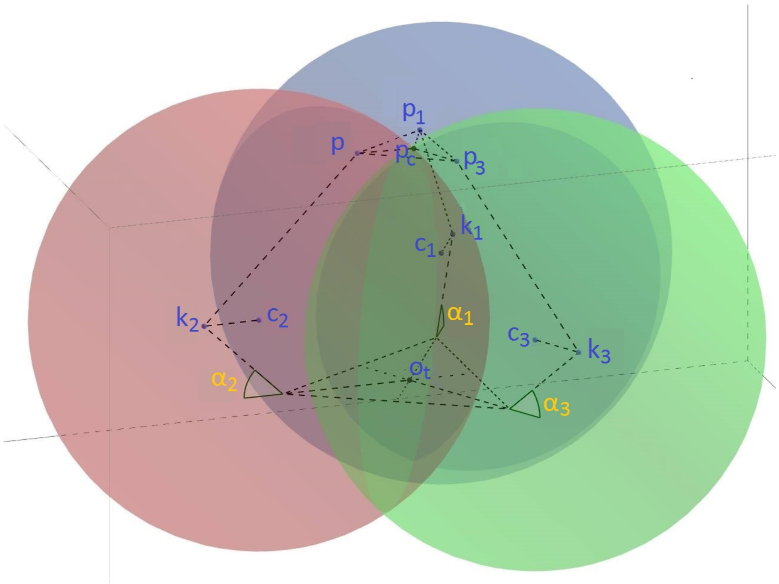


**Figure 3.3:** Tool Center Point $\boldsymbol{p}_c^t$ described by three spheres

where

$$\|\mathbf{c}_i^{t_i}\|_2^2 = l_a^2 + a^2 - 2al_a \cos\alpha_i. \tag{3.12}$$

Eq. (3.11) is then expanded for all three joints in the individual $\{t_i\}$-frames

$$\begin{aligned}
&\|\mathbf{p}_c^{t_1}\|_2^2 + \|\mathbf{c}_1^{t_1}\|_2^2 - l_r^2 - 2x_{c1}l_a \sin\alpha_1 + (\sqrt{3}y_{c1} - z_{c1})(a - l_a\cos\alpha_1) = 0,\\
&\|\mathbf{p}_c^{t_2}\|_2^2 + \|\mathbf{c}_2^{t_2}\|_2^2 - l_r^2 - 2x_{c2}l_a \sin\alpha_2 + 2z_{c2}(a - l_a\cos\alpha_2) = 0,\\
&\|\mathbf{p}_c^{t_3}\|_2^2 + \|\mathbf{c}_3^{t_3}\|_2^2 - l_r^2 - 2x_{c3}l_a \sin\alpha_3 - (\sqrt{3}y_{c3} + z_{c3})(a - l_a\cos\alpha_3) = 0,
\end{aligned} \tag{3.13}$$

where all three are in the form

$$e_i \cos\alpha_i + f_i \sin\alpha_i + g_i = 0 \quad \forall i \in \{1, 2, 3\}, \tag{3.14}$$

with the different variables found as

$$\begin{aligned}
f_1 &= f_2 = f_3 = -2x_{ci}l_a,\\
e_1 &= (-\sqrt{3}y_{c1} + z_{c1} - 2a)l_a,\\
e_2 &= -2(z_{c2} + a)l_a,\\
e_3 &= (\sqrt{3}y_{c3} + z_{c3} + 2a)l_a,\\
g_1 &= \|\mathbf{p}_c^{t_1}\|_2^2 + l_a^2 + a^2 - l_r^2 + (\sqrt{3}y_{c1} - z_{c1})a,\\
g_2 &= \|\mathbf{p}_c^{t_2}\|_2^2 + l_a^2 + a^2 - l_r^2 + 2z_{c2}a,\\
g_3 &= \|\mathbf{p}_c^{t_3}\|_2^2 + l_a^2 + a^2 - l_r^2 - (\sqrt{3}y_{c3} + z_{c3})a.
\end{aligned} \tag{3.15}$$

Tangent Half-Angle Substitution [21] is a method that can be used to solve (3.14). By defining a variable $\gamma_i \equiv \tan(\frac{\alpha_i}{2})$, $\cos\alpha_i$ and $\sin\alpha_i$ can be substituted with

$$\cos\alpha_i = \frac{1-\gamma_i^2}{1+\gamma_i^2}, \text{ and } \sin\alpha_i = \frac{2\gamma_i}{1+\gamma_i^2}. \tag{3.16}$$

Inserting (3.16) into (3.14) leads to

$$(g_i - e_i)\gamma_i^2 + (2f_i)\gamma_i + (g_i + e_i) = 0. \tag{3.17}$$

One can recognize (3.17) as a second order polynomial which can be solved with the quadratic formula

$$\gamma_{i_{1,2}} = \frac{-f_i \pm \sqrt{e_i^2 + f_i^2 - g_i^2}}{g_i - e_i} \tag{3.18}$$

This yields two solutions for $\gamma_i$, and two solutions for $\alpha_i$, since $\alpha_i = 2\arctan(\gamma_i)$. As long as the TCP is inside the workspace, the solutions are well-defined and real. An example of two different real solutions can be seen in Figure 3.4, with the knee pointing outwards (left) and with the knee pointing inwards (right). As long as the two solutions are different, this will in total yield eight different configurations for the TAC, but the solution chosen should be the one with all knees pointing outward, i.e. satisfying $|\alpha_i| < \frac{\pi}{2}$. The IPK solution can be formed into a function such that

$$\boldsymbol{\alpha} = IPK(\mathbf{p}_c^t). \tag{3.19}$$

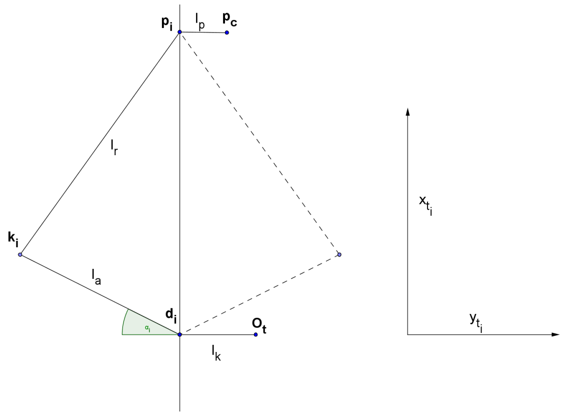


**Figure 3.4:** IPK solutions for $\boldsymbol{\alpha}$ using Tangent Half-Angle Substitution

### 3.1.3 Forward Position Kinematics

The Forward Position Kinematics (FPK) solution is the inverse of the IPK as it yields $\mathbf{p}_c^t$ given $\boldsymbol{\alpha}$. It can be solved analytically with Gaussian Elimination [5]. Eq. (3.9) can be rewritten as

$$(\boldsymbol{p}_c^t)^T \boldsymbol{p}_c^t - 2(\boldsymbol{p}_c^t)^T \boldsymbol{c}_i^t + (\boldsymbol{c}_i^t)^T \boldsymbol{c}_i^t = l_r^2. \tag{3.20}$$

One want to remove the quadratic parts of the equation to make it easier, thus lower computation cost. This can be done by introducing the variables

$$r = (\boldsymbol{p}_c^t)^T \boldsymbol{p}_c^t, \quad b_i = (\boldsymbol{c}_i^t)^T \boldsymbol{c}_i^t - l_r^2, \tag{3.21}$$

which gives

$$(\mathbf{p}_c^t)^T \mathbf{c}_i^t = (r + b_i)/2, \quad \forall i \in \{1, 2, 3\}. \tag{3.22}$$

By changing the notation to

$$\mathbf{C} = \begin{bmatrix}\mathbf{c}_1^t & \mathbf{c}_2^t & \mathbf{c}_3^t\end{bmatrix}, \quad \mathbf{1} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^T, \quad \mathbf{b} = \begin{bmatrix}b_1 & b_2 & b_3\end{bmatrix}^T, \tag{3.23}$$

and introduce the substitution variables

$$\mathbf{u} = \mathbf{C}^{-T}\mathbf{1}, \quad \mathbf{v} = \mathbf{C}^{-T}\mathbf{b}, \tag{3.24}$$

a solution for $\boldsymbol{p}_c^t$ appears as

$$\boldsymbol{p}_c^t = (r\mathbf{u} + \mathbf{v})/2. \tag{3.25}$$

A solution for $\boldsymbol{p}_c^t$ is now presented, but $r$ is still unknown. $r$ can be obtained by inserting (3.25) into (3.21)

$$r = (\boldsymbol{p}_c^t)^T \boldsymbol{p}_c^t = \frac{1}{4}(r\mathbf{u} + \mathbf{v})^T(r\mathbf{u} + \mathbf{v}). \tag{3.26}$$

This can again be rewritten to

$$(\mathbf{u}^T\mathbf{u})r^2 + (2\mathbf{u}^T\mathbf{v} - 4)r + \mathbf{v}^T\mathbf{v} = 0, \tag{3.27}$$

which can be recognized as a second order polynomial and be solved as

$$r = \frac{2 - \mathbf{u}^T\mathbf{v} \pm \sqrt{(2 - \mathbf{u}^T\mathbf{v})^2 - (\mathbf{u}^T\mathbf{u})(\mathbf{v}^T\mathbf{v})}}{\mathbf{u}^T\mathbf{u}} \tag{3.28}$$

Inserting (3.28) in (3.25) results in two analytical solutions for $\boldsymbol{p}_c^t$ given $\boldsymbol{\alpha}$. To understand why there are two solutions one can look back at Figure 3.3 where it can be observed that the three spheres crosses each other in two points. One solution is the crossing above the TAC base plate, the other is beneath. The valid solution for the crane head is the one over the TAC base plate i.e satisfying $x_c^t > x_i^t$, $\forall i \in \{1, 2, 3\}$.

### 3.1.4 Velocity Kinematics

The relationship between the velocity of the TCP, $\dot{\mathbf{p}}_c$, and the joint velocities, $\dot{\boldsymbol{\alpha}}$, can be useful for designing a control system [4, 18]. This kinematic relationship can be found by differentiating (3.9) as

$$(\mathbf{s}_i^t)^T \dot{\mathbf{s}}_i^t = 0 \tag{3.29}$$

where $\dot{\mathbf{s}}_i$ can be found as

$$\dot{\mathbf{s}_i^t} = \dot{\mathbf{p}}_c^t - \mathbf{d}_i^t \dot{\alpha}_i, \tag{3.30}$$

with

$$\mathbf{d}_i^t = \mathbf{R}_{t_i}^t \begin{bmatrix} l_a \cos\alpha_i \\ l_a \sin\alpha_i \\ 0 \end{bmatrix}. \tag{3.31}$$

Inserting (3.29) into (3.30) yields

$$(\mathbf{s}_i^t)^T(\dot{\mathbf{p}}_c^t - \mathbf{d}_i^t \dot{\alpha}_i) = 0. \tag{3.32}$$

Expanding (3.32) for $i \in \{1, 2, 3\}$ gives

$$\begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix} \dot{\mathbf{p}}_c - \begin{bmatrix} (\mathbf{s}_1^t)^T\mathbf{d}_1^t & 0 & 0 \\ 0 & (\mathbf{s}_2^t)^T\mathbf{d}_2^t & 0 \\ 0 & 0 & (\mathbf{s}_3^t)^T\mathbf{d}_3^t \end{bmatrix} \dot{\boldsymbol{\alpha}} = 0, \tag{3.33}$$

which finally becomes

$$\dot{\mathbf{p}}_c^t = \mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha})\dot{\boldsymbol{\alpha}} \tag{3.34}$$

where

$$\mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha}) = \begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix}^{-1} \begin{bmatrix} (\mathbf{s}_1^t)^T\mathbf{d}_1^t & 0 & 0 \\ 0 & (\mathbf{s}_2^t)^T\mathbf{d}_2^t & 0 \\ 0 & 0 & (\mathbf{s}_3^t)^T\mathbf{d}_3^t \end{bmatrix}, \tag{3.35}$$

is the Jacobian of the system.

### 3.1.5 Acceleration Relationship

The acceleration relationship between $\ddot{\mathbf{p}}_c$ and $\ddot{\boldsymbol{\alpha}}$ can be found by differentiating the velocity kinematics [4]. Differentiating (3.30) gives

$$\begin{gathered} (\dot{\mathbf{s}}_i^t)^T(\dot{\mathbf{p}}_c - \mathbf{d}_i\dot{\alpha}_i) + (\mathbf{s}_i)^T(\ddot{\mathbf{p}}_c - \dot{\mathbf{d}}_i\dot{\alpha}_i - \mathbf{d}_i\ddot{\alpha}_i) = 0 \quad \Rightarrow \\ (\dot{\mathbf{s}}_i^t)^T\ddot{\mathbf{p}}_c = ((\mathbf{s}_i)^T\dot{\mathbf{d}}_i + (\dot{\mathbf{s}}_i)^T\mathbf{d}_i)\dot{\alpha}_i - (\dot{\mathbf{s}}_i)^T\dot{\mathbf{p}}_c + (\mathbf{s}_i)^T\mathbf{d}_i\ddot{\alpha}_i, \end{gathered} \tag{3.36}$$

where

$$\dot{\mathbf{d}}_i = \mathbf{R}_{t_i}^t \begin{bmatrix} -l_a \sin\alpha_i \\ l_a \cos\alpha_i \\ 0 \end{bmatrix} \dot{\alpha}_i. \tag{3.37}$$

Expanding (3.36) for $i \in \{1, 2, 3\}$ yields

$$\begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix} \ddot{\mathbf{p}}_c = \begin{bmatrix} \mathbf{s}_1^t\dot{\mathbf{d}}_1 + \dot{\mathbf{s}}_1^t\mathbf{d}_1 & 0 & 0 \\ 0 & \mathbf{s}_2^t\dot{\mathbf{d}}_2 + \dot{\mathbf{s}}_2^t\mathbf{d}_2 & 0 \\ 0 & 0 & \mathbf{s}_3^t\dot{\mathbf{d}}_3 + \dot{\mathbf{s}}_3^t\mathbf{d}_3 \end{bmatrix} \dot{\boldsymbol{\alpha}} - \begin{bmatrix} (\dot{\mathbf{s}}_1^t)^T \\ (\dot{\mathbf{s}}_2^t)^T \\ (\dot{\mathbf{s}}_3^t)^T \end{bmatrix} \dot{\mathbf{p}}_c + \begin{bmatrix} \mathbf{s}_1^t\mathbf{d}_1 & 0 & 0 \\ 0 & \mathbf{s}_2^t\mathbf{d}_2 & 0 \\ 0 & 0 & \mathbf{s}_3^t\mathbf{b}_3 \end{bmatrix} \ddot{\boldsymbol{\alpha}}, \tag{3.38}$$

and since $\dot{\mathbf{p}}_c^t = \mathbf{J}\dot{\boldsymbol{\alpha}}$, (3.38) can be simplified to

$$\ddot{\mathbf{p}}_c = \begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix}^{-1} \left( \begin{bmatrix} \mathbf{s}_1^t\dot{\mathbf{d}}_1 + \dot{\mathbf{s}}_1^t\mathbf{d}_1 & 0 & 0 \\ 0 & \mathbf{s}_2^t\dot{\mathbf{d}}_2 + \dot{\mathbf{s}}_2^t\mathbf{d}_2 & 0 \\ 0 & 0 & \mathbf{s}_3^t\dot{\mathbf{d}}_3 + \dot{\mathbf{s}}_3^t\mathbf{d}_3 \end{bmatrix} - \begin{bmatrix} (\dot{\mathbf{s}}_1^t)^T \\ (\dot{\mathbf{s}}_2^t)^T \\ (\dot{\mathbf{s}}_3^t)^T \end{bmatrix} \mathbf{J} \right) \dot{\boldsymbol{\alpha}} + \mathbf{J}\ddot{\boldsymbol{\alpha}}. \tag{3.39}$$

The resulting equation is the relationship between the acceleration of the joints and the acceleration of end-effector, described as such:

$$\ddot{\mathbf{p}}_c^t = \dot{\mathbf{J}}\dot{\boldsymbol{\alpha}} + \mathbf{J}\ddot{\boldsymbol{\alpha}} \tag{3.40}$$

This relationship can be useful for creating a controller that takes the acceleration into consideration, for example to make sure actuators are not accelerated faster than the system can withstand.

### 3.1.6 Workspace

The IPK solution described in Section 3.1.2 will face difficulties if the position is outside of the reachable area of the TAC, i.e. the workspace. To this end, a check to see if the desired position is inside the workspace is necessary. Since the TAC is to be used for motion compensation, it is imperative for it to be reliable. If a desired position is found to be outside of the workspace, either other joints in the crane's body can be utilized for compensation, the craning operations can be aborted, or reduced accuracy of the motion compensation must be accepted. A method for spanning the feasible workspace can be found with the help of [1]. It is also possible to design a Delta robot based on workspace requirements [19], but this will not be discussed further in this thesis.

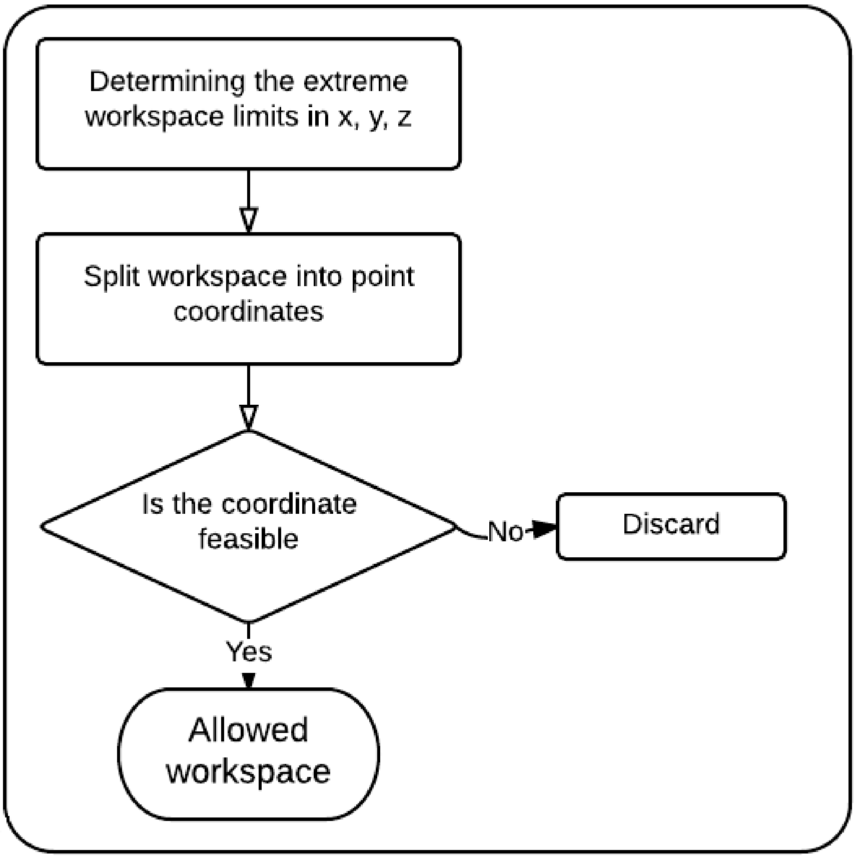

Determining the extreme workspace limits in x, y, z
Split workspace into point coordinates
Is the coordinate feasible
No
Discard
Yes
Allowed workspace

**Figure 3.5:** Block diagram for the workspace determination for the TAC

The first step in finding the workspace consists of constructing a cube containing the limits of the maximum reach of the TCP. The limits approximated stay true to the notion that the top plate can not go through the bottom plate, nor that it can be flipped, and thus these limits are found as

$$\begin{aligned} X_{max} &= l_a + l_r \\ X_{min} &= 0 \\ Y_{max} \cong Z_{max} &= l_r + l_p - l_k \\ Y_{min} \cong Z_{min} &= -(l_r + l_p - l_k). \end{aligned} \tag{3.41}$$

**Table 3.2:** Workspace example parameters

| **Parameter** | **Value** | Unit |
|---|---|---|
| $l_k$ | 4 | m |
| $l_a$ | 6.5 | m |
| $l_r$ | 14.0 | m |
| $l_p$ | 2.5 | m |
| $\alpha_{max}$ | 70 | deg |
| $\alpha_{min}$ | -40 | deg |
| $h$ | 3 | m |

The limits can be combined to form the intervals that contain both feasible and infeasible solutions for $\mathbf{p}_c^t$. These intervals are then split into individual points in three dimensions that can be tested as candidate positions for the TCP. The number of points selected will depend on the spational discretization distance $h$, which acts as the distance between points in one dimensions. This yields the vectors

$$\begin{aligned}\mathbf{x_w} &= [x_{w1}, x_{w2}, ..., x_{wn}]\\ \mathbf{y_w} &= [y_{w1}, y_{w2}, ..., y_{wn}]\\ \mathbf{z_w} &= [z_{w1}, z_{w2}, ..., z_{wn}]\end{aligned} \tag{3.42}$$

where $x_{w1} = X_{min}$, $x_{wn} = X_{max}$, $y_{w1} = Y_{min}$, $y_{wn} = Y_{max}$, $z_{w1} = Z_{min}$ and $z_{wn} = Z_{max}$ with fixed step size $h$.

In step three, all the different positions found in these vectors are then checked by testing if the angles output by IPK solution is real, and if they are inside the design range, i.e.

$$\alpha_{min} < \alpha_i < \alpha_{max} \forall i \in \{1, 2, 3\}. \tag{3.43}$$

The positions that are deemed plausible is kept and serves as the basis for the approximated workspace. The smaller $h$ chosen, the higher resolution of the workspace becomes, at the cost of computational load.

Step four consist of using the plausible coordinates on the outskirts of the workspace as the limits of the workspace, and by finding the convex hull of all the allowed coordinates, a continuous limit for the workspace is found.

With the limits found, it is possible to check if the desired position is outside the limits while the TAC is running. Appropriate actions for transgressing the limits can be to move the set point back towards the limits (saturating), involving the rest of the crane in the compensation action, or aborting the operation, among other alternatives. As it is a cumbersome process to find the limits, it should be done as an initializing process, or even done separately and merely entering the limits as parameters to the TAC's controller. A block diagram of this method can be seen in Figure 3.5.

A demonstration of the workspace check, using the parameters in Table 3.2, is plotted in Figure 3.6, 3.7, 3.8 and 3.9.

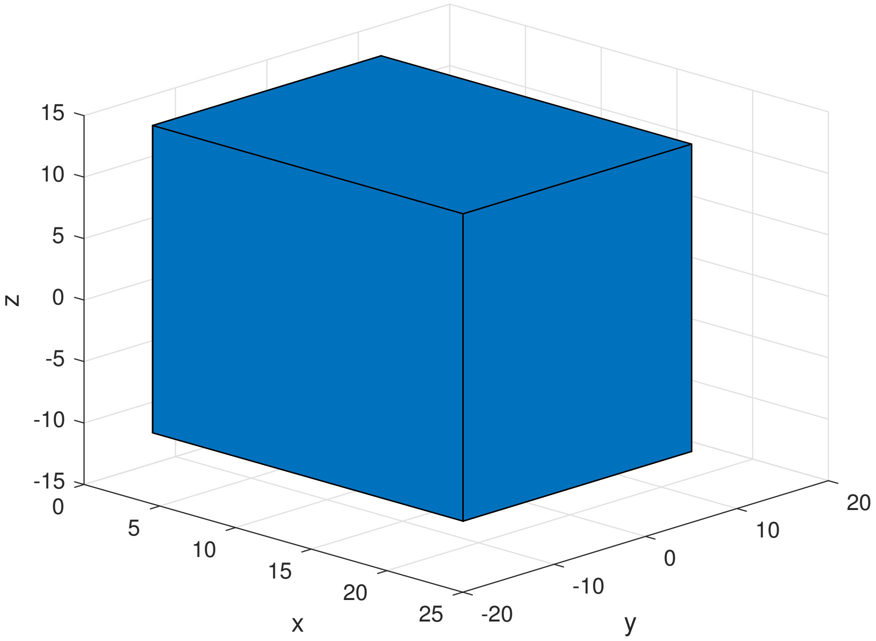


**Figure 3.6:** Step one of four to find the workspace

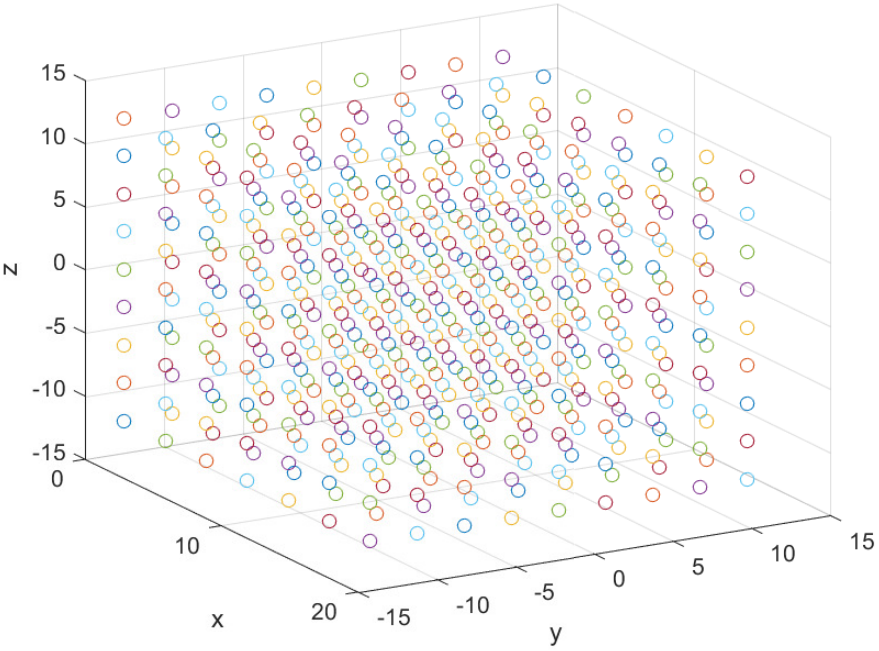


**Figure 3.7:** Step two of four to find the workspace

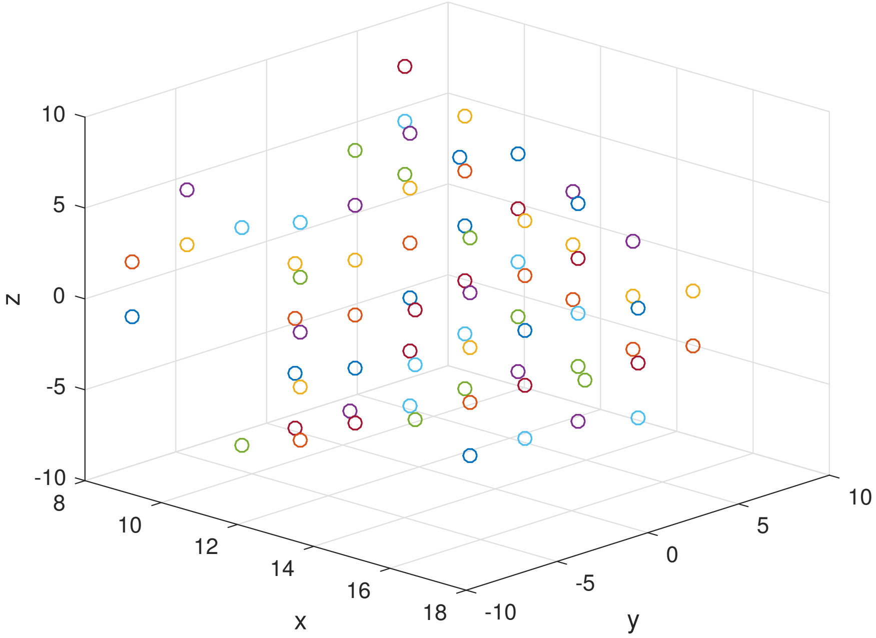


**Figure 3.8:** Step three of four to find the workspace

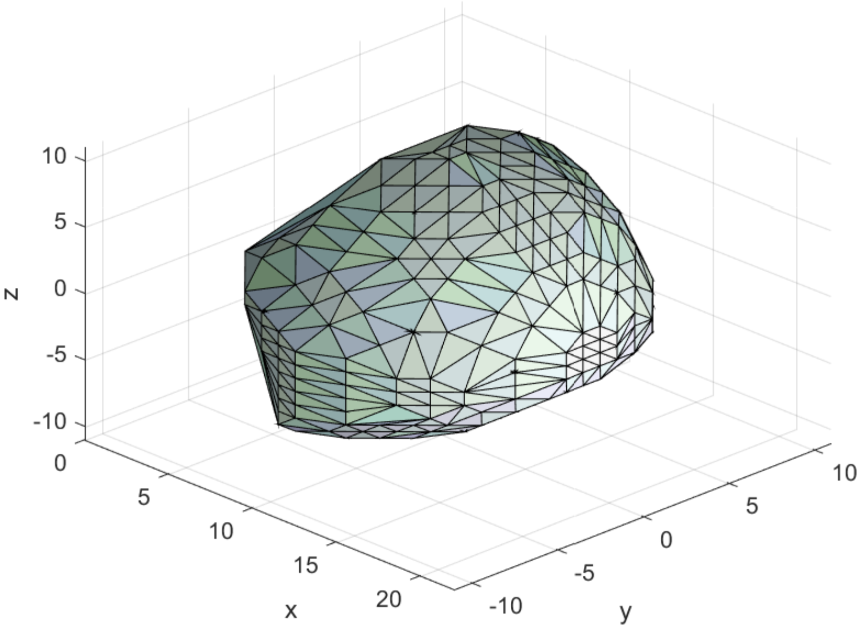


**Figure 3.9:** Step four four to find the workspace

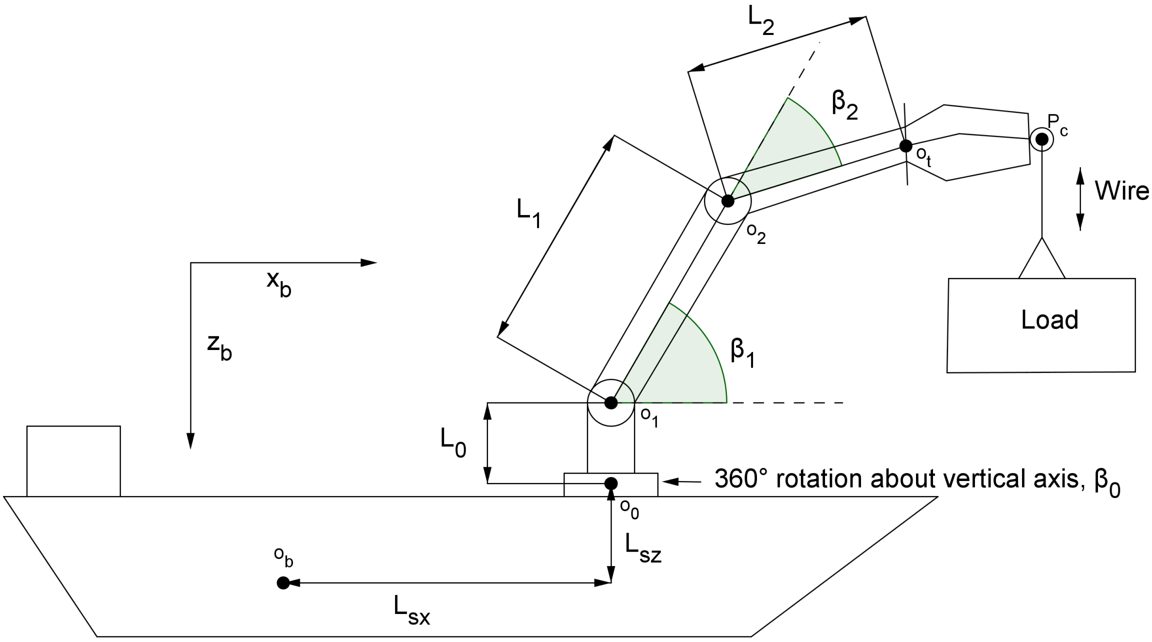


**Figure 3.10:** Ship with full crane system: an elbow manipulator and the TAC

## 3.2 Full Crane System

The full crane system is modelled as an elbow manipulator with the TAC as its end-effector, placed along the central line on the deck of a ship, as seen in Figure 3.10. The ship itself is the subject of the waves, wind and current the sea can muster, and the effects of this on the crane are important to accurately describe to be able to compensate for it. In this section the kinematics of the crane system from the origin of the ship, $\mathbf{o}_b$, to the origin of the TAC, $\mathbf{o}_t$, will be described.

### 3.2.1 Reference Frames

As described in section 2.2.1, the body-fixed reference is the {b}-frame, and the NED frame is the {n}-frame. The transformation from {b}-frame to {n}-frame can be found as

$$\boldsymbol{T}_b^n = \begin{bmatrix} \boldsymbol{R}_b^n & \boldsymbol{o}_b^n \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.44}$$

where

$$\begin{aligned} \boldsymbol{R}_b^n &= \boldsymbol{R}_{x,\phi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{z,\psi}, \\ \boldsymbol{o}_b^n &= \begin{bmatrix} x & y & z \end{bmatrix}^T, \end{aligned} \tag{3.45}$$

using the DOF's as described in section 2.2.1. Hence position and attitude of the ship can be derived if the measurements from the IMU are accurate.

### 3.2.2 Forward Position Kinematics

The forward kinematics of the full crane system can, with the knowledge of the crane's dimensions and configuration, be found by using the DH-convention [18]. With the information available in

**Table 3.3:** DH table for full crane system

| **Link** | $\boldsymbol{a}_j$ | $\boldsymbol{\alpha}_j$ | $\boldsymbol{d}_j$ | $\boldsymbol{\theta}_j$ |
|---|---|---|---|---|
| 1 | $L_{sx}$ | $\pi$ | $-L_{sz}$ | 0 |
| 2 | 0 | $\frac{\pi}{2}$ | $L_0$ | $\beta_0^*$ |
| 3 | $L_1$ | 0 | 0 | $\beta_1^*$ |
| 4 | $L_2$ | $-\frac{\pi}{2}$ | 0 | $\beta_2^*$ |

Figure 3.10, a DH-table can be made, as seen in Table 3.3. $\beta_1^*$, $\beta_2^*$ and $\beta_3^*$ are the controllable angles of the crane's three joints. Each row in the DH-table is inserted into (3.46) to garner the transformation matrices

$$\begin{gathered}\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 & 0 & L_{sx} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -L_{sz} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} c_0 & 0 & s_0 & 0 \\ s_0 & 0 & -c_0 & 0 \\ 0 & 1 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \boldsymbol{A}_3 = \begin{bmatrix} c_1 & -s_1 & 0 & L_1c_1 \\ s_1 & c_1 & 0 & L_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and } \boldsymbol{A}_4 = \begin{bmatrix} c_2 & 0 & -s_2 & L_2c_2 \\ s_2 & 0 & c_2 & L_2s_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{gathered} \tag{3.46}$$

For this crane, the product of all the transformation matrices represent the transformation from the {t}-frame to the {b}-frame

$$\boldsymbol{T}_t^b = \boldsymbol{T}_4^1 = \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_3\boldsymbol{A}_4 = \begin{bmatrix} \boldsymbol{R}_t^b & \boldsymbol{o}_t^b \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{3.47}$$

### 3.2.3 Inverse Position Kinematics

The inverse kinematics for the full crane is the method of finding the joint angles, $\boldsymbol{\beta}$, knowing the position of the end-effector, $\boldsymbol{o}_t^b$. By transforming this to the position of the first joint, $\boldsymbol{o}_t^0$, found as

$$\boldsymbol{o}_t^0 = \boldsymbol{R}_b^0\boldsymbol{o}_t^b + \boldsymbol{o}_b^0 = \begin{bmatrix} x_b^0 & y_b^0 & z_b^0 \end{bmatrix}, \tag{3.48}$$

and then from the first joint and up, the crane can be recognized as the elbow manipulator from [18]. The angles of this configuration can be found as

$$\begin{gathered}\beta_0 = \arctan\left(\frac{y_b^0}{x_b^0}\right), \quad \beta_2 = \arctan\left(\frac{\pm\sqrt{1-\delta^2}}{\delta}\right), \\ \beta_1 = \arctan\left(\frac{z_b^0 - L_0}{\sqrt{(x_b^0)^2 + (y_b^0)^2}}\right) - \arctan\left(\frac{L_2\sin\beta_2}{L_1 + L_2\cos\beta_2}\right), \end{gathered} \tag{3.49}$$

where

$$\delta = \frac{(x_b^0)^2 + (y_b^0)^2 + (z_b^0 - L_0)^2 - L_1^2 - L_2^2}{2L_1L_2}. \tag{3.50}$$

The two separate solutions for $\beta_2$ gives elbow-down or elbow-up configuration of the manipulator, as the positive or negative square root, respectively. For the configuration showcased in Figure 3.10, elbow-up is desired.

### 3.2.4 Forward Velocity Kinematics

This section will describe the forward velocity kinematics of a full crane system, from the Center of Origin (CO) to the TAC. The methods used in this section are explained more generally in Section 2.1.4. The velocity kinematics of the full crane can be described by finding the Jacobian matrix, $\mathbf{J}_\beta$, which describes the velocity of the end effector, $\boldsymbol{\xi}$, with the joint velocities, $\dot{\boldsymbol{\beta}} = \begin{bmatrix}\dot{\beta}_0 & \dot{\beta}_1 & \dot{\beta}_2\end{bmatrix}^T$. The velocity kinematics can be found as

$$\boldsymbol{\xi}_t^b = \begin{bmatrix}\boldsymbol{v}_t^b \\ \boldsymbol{\omega}_t^b\end{bmatrix} = \mathbf{J}_\beta \dot{\boldsymbol{\beta}}. \tag{3.51}$$

Where the $\mathbf{J}_\beta$ can be found by using (2.7), and since the first link in Table 3.3 consist of fixed lengths without any variable movement, it is skipped and the Jacobian matrix starts from the second link in Table 3.3, and can be found as

$$\mathbf{J}_\beta \begin{bmatrix}\mathbf{J}_{v\beta} \\ \mathbf{J}_{\omega\beta}\end{bmatrix} = \begin{bmatrix}\mathbf{z}_1^0 \times (\mathbf{o}_4^0 - \mathbf{o}_1^0) & \mathbf{z}_2^0 \times (\mathbf{o}_4^0 - \mathbf{o}_2^0) & \mathbf{z}_3^0 \times (\mathbf{o}_4^0 - \mathbf{o}_3^0) \\ \mathbf{z}_1^0 & \mathbf{z}_2^0 & \mathbf{z}_3^0\end{bmatrix}, \tag{3.52}$$

where all vectors can be calculated using the matrices in (3.46).

The methods derived in this chapter will be used in the following chapters.

# Chapter 4
# Crane and Vessel Simulator

This chapter will describe the simulator created in MATLAB/Simulink to verify and explore the crane system. The simulator consist of a full sized supply ship, with a the full crane described in Chapter 3 on it's deck. The simulator uses a vessel simulator and wave generator from the MSS toolbox [8] to garner the DOF's og the ship.

## 4.1 Simulator Setup

The TAC is to be controlled by four motors which manipulate $\boldsymbol{\alpha}$, the angles of the arms, and in the simulator the motors are modeled as the first order system

$$\frac{\boldsymbol{\alpha}}{\mathbf{u}_\alpha}(s) = \frac{K_{DC}}{T_{DC}s + 1}, \tag{4.1}$$

where values for the motor constants $K_{DC}$ and $T_{DC}$ can be found in Table 4.1. Inverse Laplace transformation gives

$$\dot{\boldsymbol{\alpha}} = -\frac{1}{T_{DC}}\boldsymbol{\alpha} + \frac{K_{DC}}{T_{DC}}\mathbf{u}. \tag{4.2}$$

To simulate the TAC itself (3.34) is used as a simplified representation of the dynamics. Inserting (4.2) into (3.34) yields

$$\dot{\mathbf{p}}_c^t = \mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha})(-\frac{1}{T_{DC}}\boldsymbol{\alpha} + \frac{K_{DC}}{T_{DC}}\mathbf{u}_\alpha). \tag{4.3}$$

which yields the TAC simulator.

The rest of the crane is created as an elbow manipulator as described in Chapter 3.2. As the focus of this thesis is mostly on the TAC, the crane controls are simplified to consist merely of the joint angles, $\boldsymbol{\beta}$, and angle speed, $\dot{\boldsymbol{\beta}}$. Essential parameter values used in the crane simulator configuration can be found in Table 4.1.

From the MSS toolbox, two blocks were used. A wave generation function which outputs the first- and second-order wave-induced forces and moments $\boldsymbol{\tau}_{wave1}$ and $\boldsymbol{\tau}_{wave2}$ as the force input for the vessel simulator. The vessel simulator itself is a "6DOF DP zero speed model with fluid memory" block, which outputs the DOF's in $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ as described in chapter 2.2.1. As this is a simplified simulator and the focus was to see if the TAC could work in perfect condition, measurement noise

**Table 4.1:** Simulator Crane Parameters

| **Parameter** | **Value** | Unit |
|---|---|---|
| $l_k$ | 2.00 | m |
| $l_a$ | 3.25 | m |
| $l_r$ | 7.00 | m |
| $l_p$ | 1.25 | m |
| $\alpha_{min}$ | -45 | deg |
| $\alpha_{max}$ | 70 | deg |
| $T_{DC}$ | 0.2 | s |
| $K_{DC}$ | 1 | - |
| $K_p$ | 10 | - |
| $K_i$ | 60 | - |
| $K_d$ | 1 | - |
| $L_{sx}$ | -20.70 | m |
| $L_{sz}$ | 3.00 | m |
| $L_0$ | 3.00 | m |
| $L_1$ | 19.20 | m |
| $L_2$ | 9.60 | m |

**Table 4.2:** Simulator Ship Parameters

| **Parameter** | **Value** | **Unit** | **Description** |
|---|---|---|---|
| Lpp | 82.8 | m | Length between perpendiculars |
| T | 6 | m | Draught |
| B | 19.2 | m | Largest width |
| m | 6.3622 | kTon | Mass |

a real IMU would most certainly have encountered was omitted, and $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are output directly as IMU measurements. The ship model was configured as a supply ship using a parameter file found in the MSS toolbox, of which the main parameters can be seen in Table 4.2. The crane has been dimensioned appropriately to the supply ship, which can be seen in Figure 4.1.

## 4.2 Simulator Control System

The desired initial position of the TCP, $\boldsymbol{p}_d^t$, should be such that it is able to compensate to the greatest effect. For simplicity it is placed in the middle of the workspace, such that the TCP has the largest freedom of movement in all directions, although this is not necessarily the best position for all scenarios. This gives a desired position in the {t}-frame of $\boldsymbol{p}_d^t = [x_d \quad 0 \quad 0]^T$ where $x_d$ can be found by analyzing the workspace. The desired position of the TCP in the {n}-frame is found as

$$\boldsymbol{p}_d^n = \boldsymbol{R}_t^b \boldsymbol{p}_d^t + \boldsymbol{o}_t^b, \tag{4.4}$$

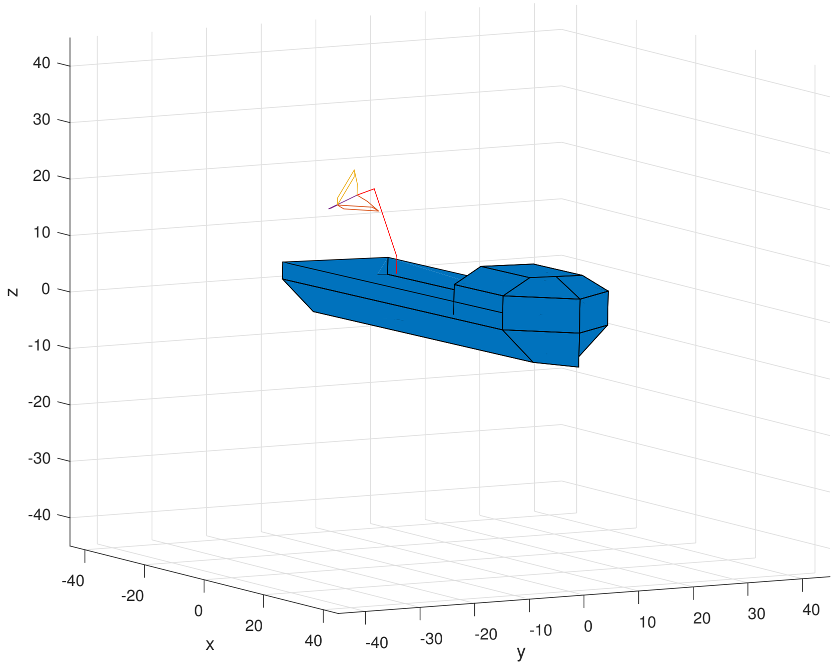


**Figure 4.1:** MATLAB visualization of full ship with crane

which is unaffected by the environmental forces. The actual TAC position in {n}-frame, $\boldsymbol{o}_t^n$, can be found as

$$\boldsymbol{o}_t^n = \boldsymbol{R}_b^n \boldsymbol{o}_t^b + \boldsymbol{o}_b^n. \tag{4.5}$$

To compensate for the vessel's motion, the TCP should be moved equally in the opposite direction of the disturbance, thus the reference should be a vector from $\boldsymbol{o}_t^t$ to $\boldsymbol{p}_d^t$, and it can be found as

$$\boldsymbol{p}_{ref}^t = \boldsymbol{R}_n^t \boldsymbol{p}_{ref}^n, \tag{4.6}$$

where $\boldsymbol{R}_n^t = (\boldsymbol{R}_t^b)^T(\boldsymbol{R}_b^n)^T$ and $\boldsymbol{p}_{ref}^n = \boldsymbol{p}_d^n - \boldsymbol{o}_t^n$. After checking if $\boldsymbol{p}_{ref}^t$ is inside the workspace, the reference angles, $\boldsymbol{\alpha}_{ref}$, can be found with the use of the IPK function found in Section 3.1.2, such that

$$\boldsymbol{\alpha}_{ref} = IPK(\boldsymbol{p}_{ref}^t). \tag{4.7}$$

The velocity reference can be found by differentiating (4.6)

$$\begin{aligned} \mathbf{v}_{ref}^t &= \dot{\mathbf{p}}_{ref}^t \\ &= \dot{\mathbf{R}}_n^t \mathbf{p}_{ref}^n + \mathbf{R}_n^t \dot{\mathbf{p}}_{ref}^n \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} \dot{\mathbf{R}}_n^t &= (\dot{\boldsymbol{R}}_t^b)^T(\boldsymbol{R}_b^n)^T + (\boldsymbol{R}_t^b)^T(\dot{\boldsymbol{R}}_b^n)^T \\ &= -(\mathbf{S}(\boldsymbol{\omega}_t^b)(\mathbf{R}_t^b)^T + (\mathbf{R}_t^b)^T\mathbf{S}(\boldsymbol{\omega}_b^n))(\mathbf{R}_b^n)^T \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} \dot{\mathbf{p}}_{ref}^n &= \dot{\mathbf{R}}_t^b \boldsymbol{p}_d^t + \dot{\mathbf{o}}_t^b - \dot{\mathbf{R}}_b^n \mathbf{o}_t^b - \mathbf{R}_b^n \dot{\mathbf{o}}_t^b - \dot{\mathbf{o}}_b^n \\ &= \mathbf{S}(\boldsymbol{\omega}_t^b)\mathbf{R}_t^b \boldsymbol{p}_d^t - \mathbf{S}(\boldsymbol{\omega}_b^n)\mathbf{R}_b^n \mathbf{o}_t^b + (\mathbf{I} - \mathbf{R}_b^n)\mathbf{v}_t^b - \mathbf{v}_b^n \end{aligned} \tag{4.10}$$

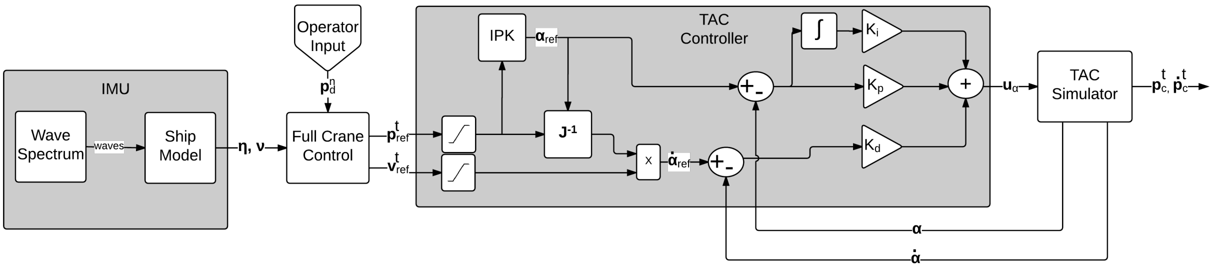


**Figure 4.2:** Block diagram representation of the simulator

with

$$\begin{aligned} \mathbf{v}_b^n &= \begin{bmatrix} u & v & w \end{bmatrix}^T, \\ \boldsymbol{\omega}_b^n &= \begin{bmatrix} p & q & r \end{bmatrix}^T, \end{aligned} \tag{4.11}$$

$\boldsymbol{\omega}_t^b$ and $\mathbf{v}_t^b$ from (3.51). The TCP velocities are translated to the joint angle velocities by use of the Jacobian matrix from Section 3.1.4,

$$\dot{\boldsymbol{\alpha}}_{ref} = \mathbf{J}^{-1}(\mathbf{p}_{ref}^t, \boldsymbol{\alpha}_{ref})\mathbf{v}_{ref}^t. \tag{4.12}$$

With error dynamics as $\tilde{\boldsymbol{\alpha}} \equiv \boldsymbol{\alpha}_{ref} - \boldsymbol{\alpha}$ and $\dot{\tilde{\boldsymbol{\alpha}}} \equiv \dot{\boldsymbol{\alpha}}_{ref} - \dot{\boldsymbol{\alpha}}$, the input to the DC-motors, $\mathbf{u}_\alpha = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$, can be chosen with the use of a PID controller as such

$$\mathbf{u}_\alpha = \mathbf{K}_p\tilde{\boldsymbol{\alpha}} + \mathbf{K}_i \int \tilde{\boldsymbol{\alpha}}(\tau)d\tau + \mathbf{K}_d\dot{\tilde{\boldsymbol{\alpha}}}, \tag{4.13}$$

where the controller gains are chosen through regulator tuning. The full simulator setup with controller can be seen in Figure 4.2.

## 4.3 Simulation Results

Simulations of sea states with significant wave height $H_s$ of up to and including 3.0 meters was conducted successfully. This puts the sea state to 5, meaning rough sea [8]. Higher sea states than this caused the TAC setpoint to be outside of the workspace. In Figure 4.3, $\boldsymbol{\eta}$ simulated by the vessel simulator during $H_s = 5$ can be seen, and it should be noted that the ship is drifting a bit in the $y$-direction. Two simulations will be shown in this section, one where the crane is fixed, and one where it is rotated about its base joint.

### 4.3.1 Simulation with Fixed Crane

In this simulation, the crane is fixed into a position such that the TAC is in alignement with the ship, and the crane is pointed to the port side of the ship, thus

$$\beta_0 = -\frac{\pi}{4} \text{ rad}, \quad \beta_1 = \beta_2 = \frac{\pi}{4} \text{ rad}, \quad \dot{\beta}_1 = \dot{\beta}_2 = \dot{\beta}_3 = 0 \text{ rad/s}, \tag{4.14}$$

Figure 4.4 compares $\mathbf{p}_c^t$ and $\mathbf{p}_{ref}^t$, while Figure 4.5 compares them in the {n}-frame, along with the uncompensated TCP position $\mathbf{p}_s^n$. Figure 4.6 shows a comparison of the total error between desired and actual position of the TCP, $e_s$ is the uncompensated, while $e_c$ is the compensated position. It can be seen from Figure 4.4 that the TCP position stays at it's reference, and from Figure 4.6 that the reference is correct in terms of motion compensation.

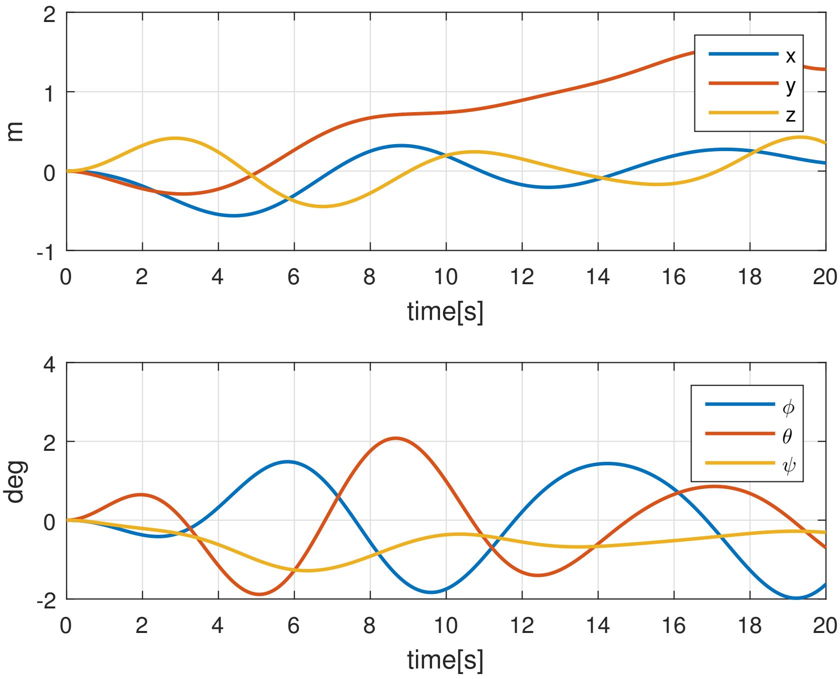


**Figure 4.3:** Vessel position and orientation $\boldsymbol{\eta}$, while crane is still

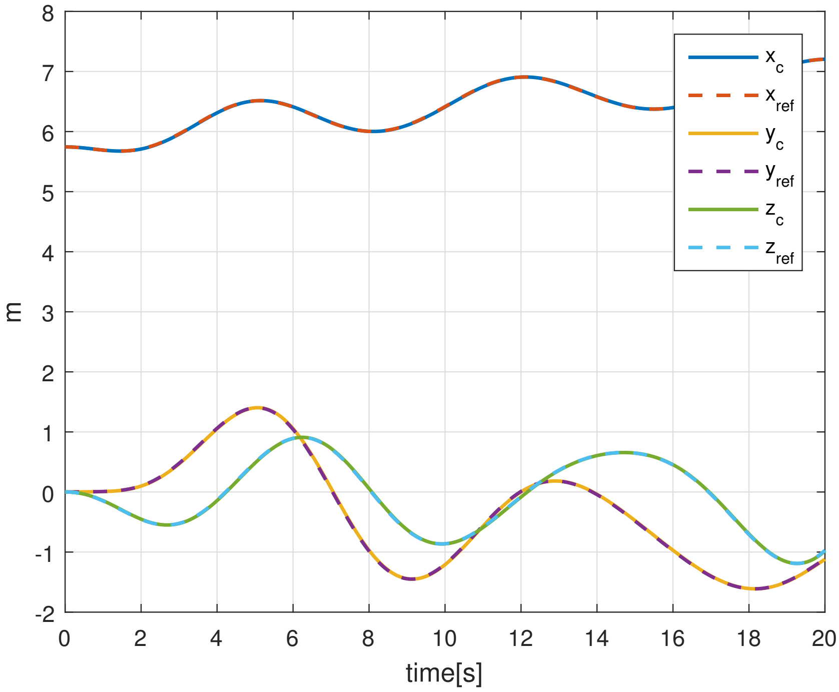


**Figure 4.4:** TCP position $\boldsymbol{p}_c^t$, while crane is fixed

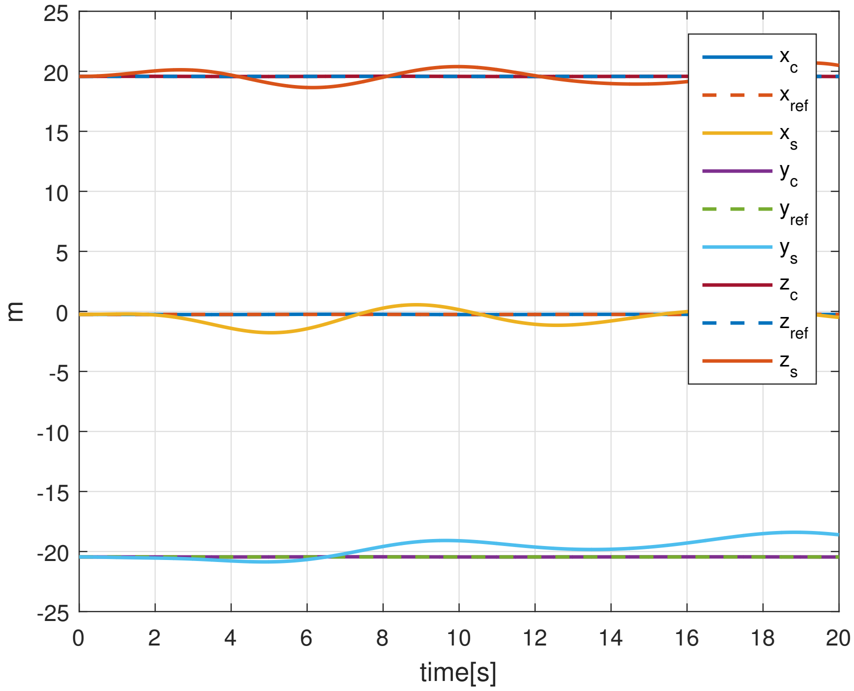


**Figure 4.5:** TCP position $\boldsymbol{p}_c^n$, while crane is fixed

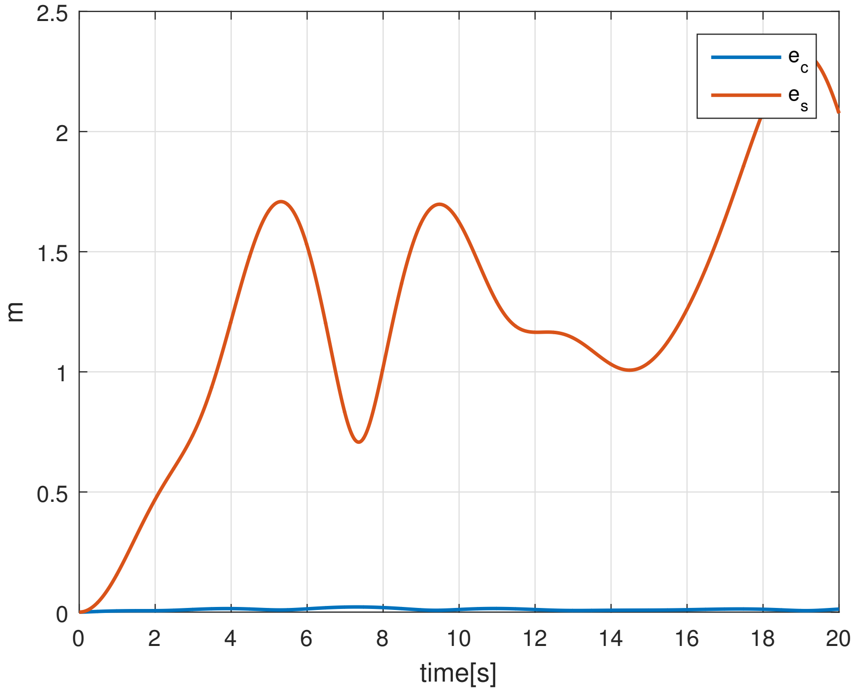


**Figure 4.6:** Comparison of position error compensated vs uncompensated, while crane is fixed

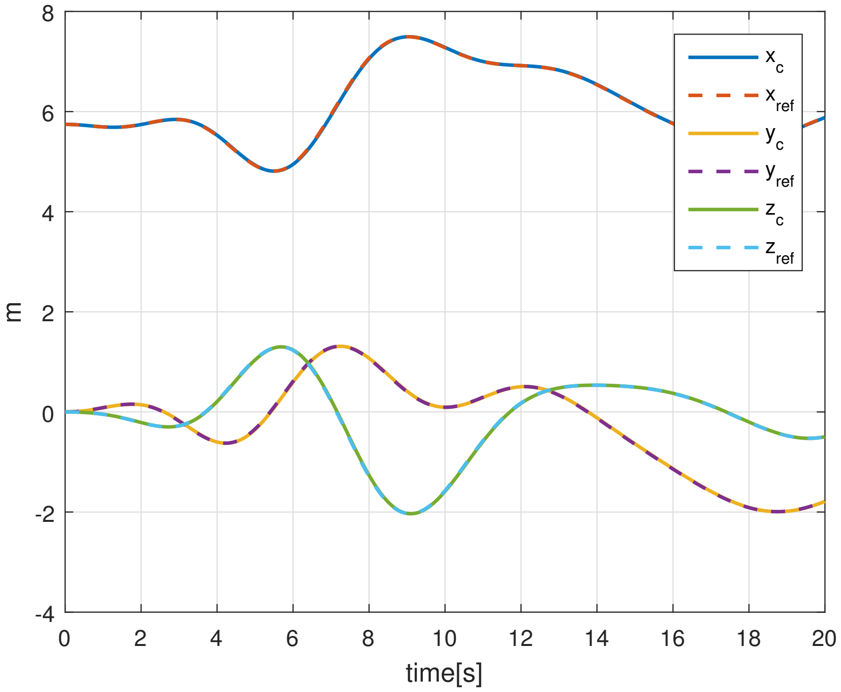


**Figure 4.7:** TCP position $\boldsymbol{p}_c^t$, while crane is moving

### 4.3.2 Simulation with Moving Crane

The moving crane simulation is done by moving the base joint angle $\beta_0$ 360° with a sinus function that is easily differentiated, thus

$$\begin{aligned}&\beta_0 = 2\pi \sin(\frac{\pi}{40}t) \text{ rad}, \quad \dot{\beta}_0 = \frac{\pi^2}{20}\cos(\frac{\pi}{40}t) \text{ rad/s},\\ &\beta_1 = \beta_2 = \frac{\pi}{4} \text{ rad}, \quad \dot{\beta}_1 = \dot{\beta}_2 = 0 \text{ rad/s},\end{aligned} \tag{4.15}$$

where $t$ is the continuous simulation time. This crane configuration has the crane moving in a circle about the ship in 20 seconds, with the TAC horizontal with the ship. Figures 4.7, 4.8 and 4.9 show the plots for the experiment. Comparing Figures 4.9 and 4.6, which consider the position error in the two cases, it is clear that they are about as good, which means that the algorithm works even as the crane is moving.

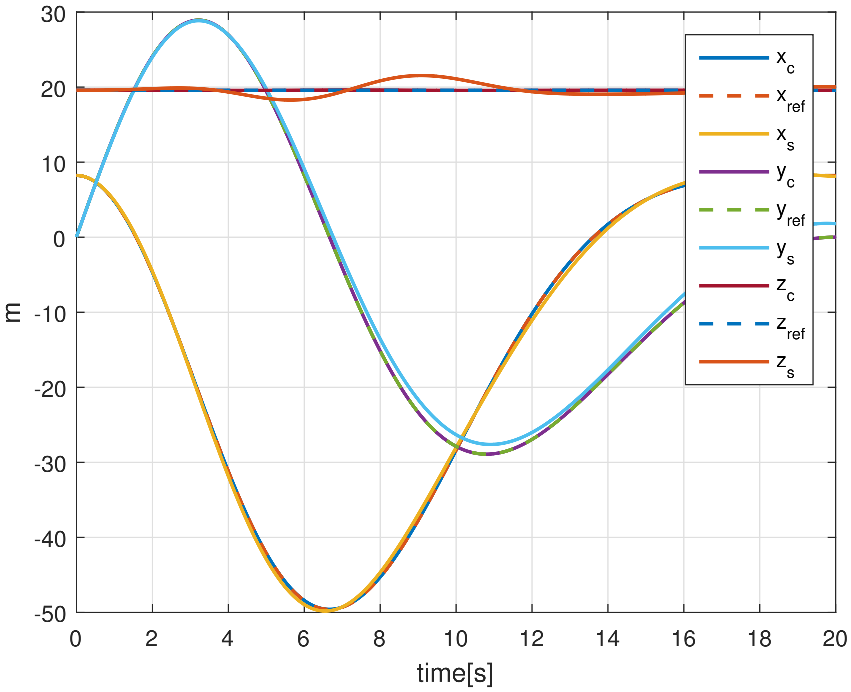


**Figure 4.8:** TCP position $\boldsymbol{p}_c^n$, while crane is moving

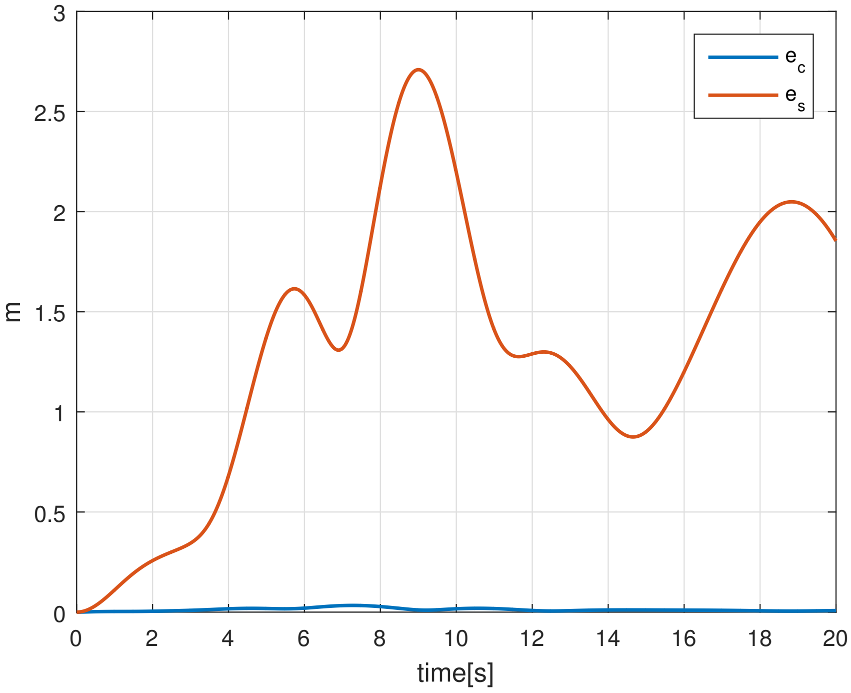
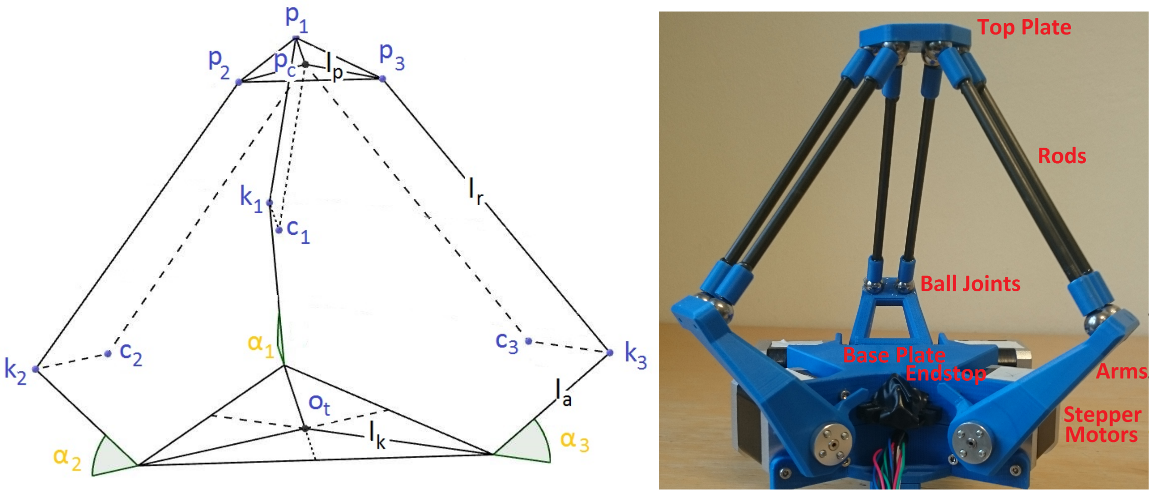


**Figure 4.9:** Comparison of position error compensated vs uncompensated, while crane is moving

# Chapter 5 TAC Prototype

The previous chapters have described the mathematical modelling of the TAC together with simulations conducted with MATLAB/Simulink. To prove the concept of motion compensation using a Delta robot and an IMU even further a TAC prototype was built. This chapter will present the signal processing and control algorithms used, as well as describe the hardware and software of the prototype. Figure 5.1 compares the geometric representation of the TAC, developed in Section 3.1, with the prototype.



**(a)** TAC Geometry

**(b)** TAC Prototype

**Figure 5.1:** Comparison of the prototype and the geometry drawing

## 5.1 Processing of IMU data

This section presents the method for processing the signals measured by the IMU. The data received from the IMU are linear acceleration in all three axis and the Euler angles, which describe the attitude. Due to inaccuracies in the measurements, vibrations and magnetic disturbances caused by the stepper motors, these signals are contaminated by noise. As the IMU is placed inside the TAC's baseplate, the vectors are given in the $\{t\}$-frame and their notation is explained in Table 5.1.

**Table 5.1:** Unfiltered and filtered IMU data vectors

| | |
|---|---|
| $\mathbf{\Theta}_x$ - measured attitude | $\mathbf{\Theta}_y$ - filtered attitude |
| $\mathbf{a}_x^t$ - measured acceleration | $\mathbf{a}_y^t$ - filtered acceleration |
| $\mathbf{v}_x^t$ - estimated velocity | $\mathbf{v}_y^t$ - filtered velocity |
| $\mathbf{p}_x^t$ - estimated position | $\mathbf{p}_y^t$ - filtered position |

The processing used to get the filtered variables used in the control algorithms can be summed up to seven steps, described in the following. Filters used are LPF and HPF Butterworth filters, described in chapter 2.3.1, and integration is done using Euler's method described in chapter 2.3.3.

1. Estimate $\mathbf{\Theta}_y^t$ is found by low pass filtering $\mathbf{\Theta}_x^t$ to suppress noise. A first order LPF with $f_c = 3$Hz was chosen for this task.
2. Remove gravity from acceleration measurements. This is done by utilizing eq. (2.9) from Section 2.2.2. The acceleration measurements are rotated from the $\{b\}$-frame to the $\{n\}$-frame, subtracting the gravitational vector and rotating back to the $\{b\}$-frame.
3. Estimate $\mathbf{a}_y^t$ by low- and high pass filtering $\mathbf{a}_x^t$ to remove noise and bias. A LPF with $f_c = 3$Hz and a HPF with $f_c = 1.5$Hz were chosen. First the measurements was run through the LPF. Then they was run through the HPF two times, this to achieve a lower output gain, -3dB for one filter and -6dB for two, at the cutoff frequency.
4. Estimate $\mathbf{v}_x^t$ by integrating $\mathbf{a}_y^t$. With the time difference, $dt$, between last integration and the filtered acceleration, the velocity is found at time $t$ as $\mathbf{v}_x^t(t) = \mathbf{v}_x^t(t-1) + dt \cdot \mathbf{a}_y^t(t)$.
5. Estimate $\mathbf{v}_y^t$ by high pass filtering $\mathbf{v}_x^t$ to remove bias. A second order HPF $f_c = 1.5$Hz was chosen. A second order filter has a steeper cutoff than a first order filter, see Section 2.3.1, which was found suitable for this task.
6. Estimate $\mathbf{p}_x^t$ by integrating $\mathbf{v}_y^t$. Same method as used to find $\mathbf{v}_x^t$, positional deviation at time $t$ is found as $\mathbf{p}_x^t(t) = \mathbf{p}_x^t(t-1) + dt \cdot \mathbf{v}_y^t(t)$.
7. Estimate $\mathbf{p}_y^t$ by high pass filtering $\mathbf{p}_x^t$ to remove bias. For this task a first order HPF with $f_c = 1.5$Hz was found sufficient.

## 5.2 Control Algorithms

This section details the two control approaches that included feedback from the IMU that were designed and tested for the prototype. The first one uses acceleration as the main source of feedback, while the other uses the attitude as the main source of feedback.

### 5.2.1 Closed Loop TAC Control with Acceleration Feedback

The control algorithm described here uses $\mathbf{p}_y^t$, $\mathbf{v}_y^t$ and $\mathbf{a}_y^t$ as the feedback to control the TAC. This PID-controller is of the form

$$\mathbf{v}_{ref} = K_p \mathbf{v}_e + K_i \int_0^t \mathbf{v}_e(\tau) d\tau + K_d \frac{d\mathbf{v}_e(t)}{dt}, \tag{5.1}$$

where $\mathbf{v}_{ref}$ is the reference velocities of the TCP and $\mathbf{v}_e$ is the velocity error. By rewriting (5.1) to

$$\mathbf{v}_{ref}^t = K_p(\mathbf{v}_d^t - \mathbf{v}_y^t) + K_i(\mathbf{p}_d^t - \mathbf{p}_y^t) + K_d(\mathbf{a}_d^t - \mathbf{a}_y^t), \tag{5.2}$$

a PID-controller for the TAC is designed . $\boldsymbol{K}_p$, $\boldsymbol{K}_i$ and $\boldsymbol{K}_d$ are the controller gain matrices and on the form

$$\mathbf{K}_p = \begin{bmatrix} K_{p,1} & 0 & 0 \\ 0 & K_{p,2} & 0 \\ 0 & 0 & K_{p,3} \end{bmatrix}, \quad \mathbf{K}_i = \begin{bmatrix} K_{i,1} & 0 & 0 \\ 0 & K_{i,2} & 0 \\ 0 & 0 & K_{i,3} \end{bmatrix}, \quad \mathbf{K}_d = \begin{bmatrix} K_{d,1} & 0 & 0 \\ 0 & K_{d,2} & 0 \\ 0 & 0 & K_{d,3} \end{bmatrix}, \tag{5.3}$$

and the controller gains were tuned to

$$\begin{aligned} &K_{p,1} = 100, \quad K_{p,2} = K_{p,3} = 150, \\ &K_{i,1} = 50, \quad K_{i,2} = K_{i,3} = 75, \\ &K_{d,1} = 500, \quad K_{d,2} = K_{d,3} = 750. \end{aligned} \tag{5.4}$$

As the objective for this algorithm is to keep a fixed position, and not follow a trajectory, the desired acceleration, velocity and position are zero. Therefore (5.2) can be rewritten to

$$\mathbf{v}_{ref}^t = -\mathbf{K}_p\mathbf{v}_y^t - \mathbf{K}_i\mathbf{p}_y^t - \mathbf{K}_d\mathbf{a}_y^t. \tag{5.5}$$

The control input for the TAC are the desired angle velocities, $\dot{\boldsymbol{\alpha}}_{ref}$, which can be calculated by

$$\dot{\boldsymbol{\alpha}}_{ref} = \mathbf{J}^{-1}(\mathbf{p}_c^t, \boldsymbol{\alpha})\mathbf{v}_{ref}^t \tag{5.6}$$

where $\mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha})$ is the Jacobian matrix of the TAC system, as described in Section 3.1.4.

### 5.2.2 Closed Loop TAC Control with Attitude Feedback

This control algorithm uses Euler angles measured by the IMU to control the TAC. As such, it will not try to compensate for translational motions, only for rotations. The desired position of the TCP, $\mathbf{p}_d^n$, should be placed in the middle of the workspace to yield maximum mobility. For the TAC, this was found to be where all stepper angles are equal and about 20°, and $\mathbf{p}_d^n$ can be found by

$$\mathbf{p}_d^n = FPK(\boldsymbol{\alpha}_d) \tag{5.7}$$

where $\boldsymbol{\alpha}_d = [20 \quad 20 \quad 20]^T$. The reference position for the TCP in the {t}-frame, $\mathbf{p}_{ref}^t$, is then found by

$$\mathbf{p}_{ref}^t = \mathbf{R}_n^t(\boldsymbol{\Theta}_y)\mathbf{p}_d^n. \tag{5.8}$$

And the desired stepper angles $\boldsymbol{\alpha}_{ref}$ can be found by using the IPK function described in chapter 3 as

$$\boldsymbol{\alpha}_{ref} = IPK(\mathbf{p}_{ref}^t). \tag{5.9}$$

The control input is found using a P-controller of the form

$$\dot{\boldsymbol{\alpha}}_{ref} = K_p(\boldsymbol{\alpha}_{ref} - \boldsymbol{\alpha}) \tag{5.10}$$

where $\boldsymbol{K}_p$ is the controller gain matrix

$$\boldsymbol{K}_p = \begin{bmatrix} K_{p,1} & 0 & 0 \\ 0 & K_{p,2} & 0 \\ 0 & 0 & K_{p,3} \end{bmatrix}, \qquad (5.11)$$

with tunes gains given as $K_{p,1} = K_{p,2} = K_{p,3} = 100$.

A block diagram explaining how the IMU signals are processed, before being used in the two controller algorithms can be found in Figure 5.2.

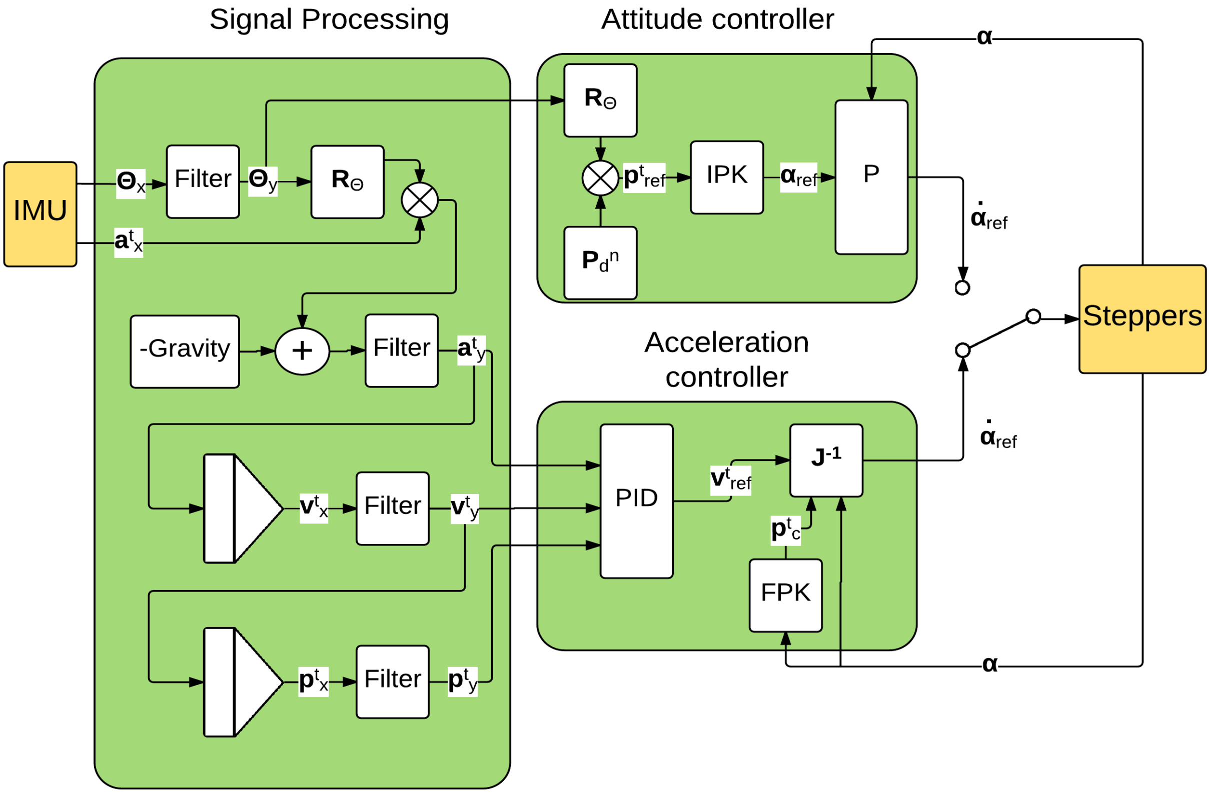


**Figure 5.2:** Block diagram showing the filtering and controller algorithm for the TAC prototype

## 5.3 Hardware

This section will provide a description of the hardware used to build the prototype together with a circuit schematic and wire list. Also described is a rig that was built as an attempt to stifle some of the noise caused by the steppers. When building the TAC prototype, a 3D-printer was used to create some components for the TAC's skeleton while others were ordered. Both a general description of the components and an explanation of how they were used in this project will be presented in the following.

### 5.3.1 Components

A complete list of the components used to build this prototype is provided in Table 5.2. Datasheets for all electrical components can be found in Appendix C.

**Table 5.2:** Prototype Components

| Reference | Part | Description | Quantity |
|---|---|---|---|
| | Top Plate | 3D-printed | 1 |
| | Lower Base Plate | 3D-printed | 1 |
| | Upper Base Plate | 3D-printed | 1 |
| | Single Motor Arms | 3D-printed | 2 |
| | Double Motor Arms | 3D-printed | 1 |
| | Handle | 3D-printed | 1 |
| | Rod Sleeves | 3D-printed | 12 |
| | Rods | Carbon Tubes | 6 |
| | Rod Magnets | - | 12 |
| | Steel Balls | Low carbon, soft steel bearings catapult ammo balls | 12 |
| | Motor shaft mounting | Universal Mounting Hub | 4 |
| -S1/S2L/S2R/S3 | Stepper Motors | NEMA 17 | 4 |
| -R1 | Control Board Kit | RAMPS 2560 Electronic | 1 |
| -R2 | Control Board | Arduino Uno | 1 |
| -I1 | IMU | 9 DOF Razor IMU, | 1 |
| | IMU Programmer | FTDI Basic Breakout | 1 |
| -E1/E2/E3 | Optical End Stop Switch | TCST2103 | 3 |
| -D1 | Computer | Running a MATLAB script | 1 |

#### Skeleton

The base plate, arms, top plate and rod sleeves were all 3D-printed, i.e. all the blue components of the TAC seen in Figures 5.1 and 5.3. The parts were drawn by Ulstein in Solidworks and 3D-printed at NTNU, using Acrylonitrile Butadiene Styrene (ABS) plastic. The Solidworks drawing of the prototype can be seen from different angles in Appendix B.

Carbon tubes was used to make the TAC rods because of their rigidity and low weight. The tubes were ordered in length of about one meter and cut into fitting sizes. The ball joints were created with magnets and steel balls. The rods were created with the carbon tubes, with a magnet on each end glued together in a rod sleeve, see Figure 5.4. Two steel balls were glued to the arms, and in total 6 were glued to the top plate, such that two rods go between each arm and the top plate, as seen in Figure 5.1. This created stepless joints such that the TCP can move smoothly in any direction.

The arms were mounted to the base plate together with the stepper motors, fastened with mounting hubs such that the stepper motion moves the arms. The top arm was designed to be controlled by

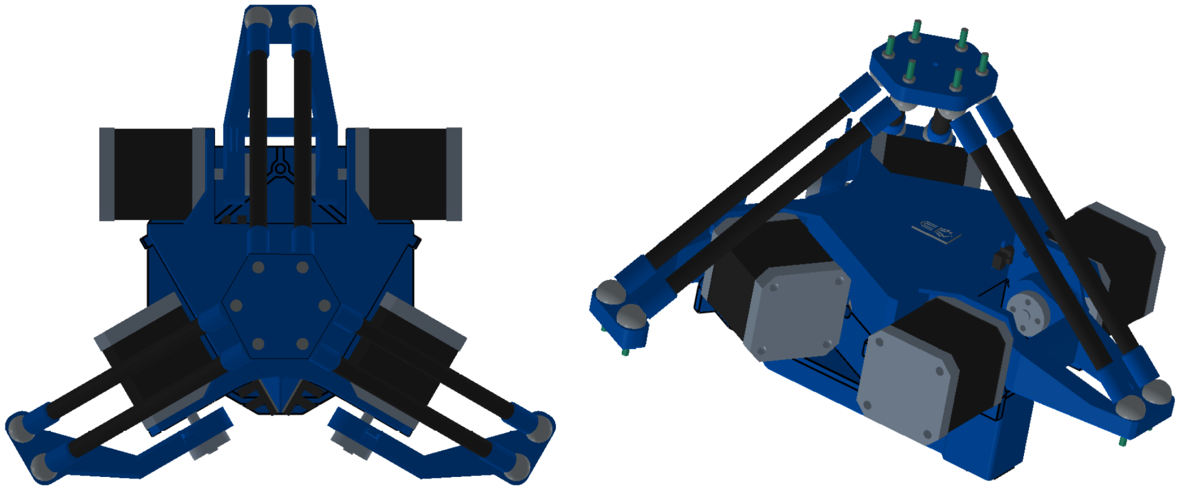


**Figure 5.3:** Prototype Solidworks drawings

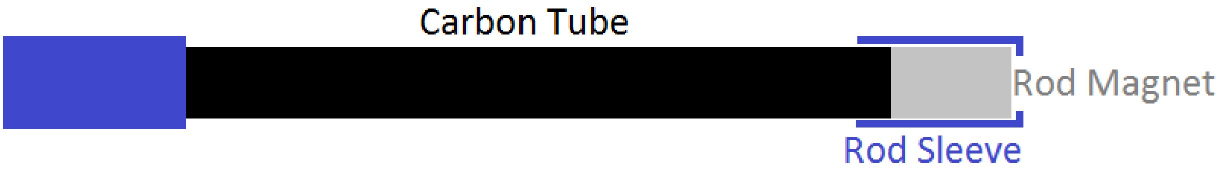


**Figure 5.4:** Rod construction

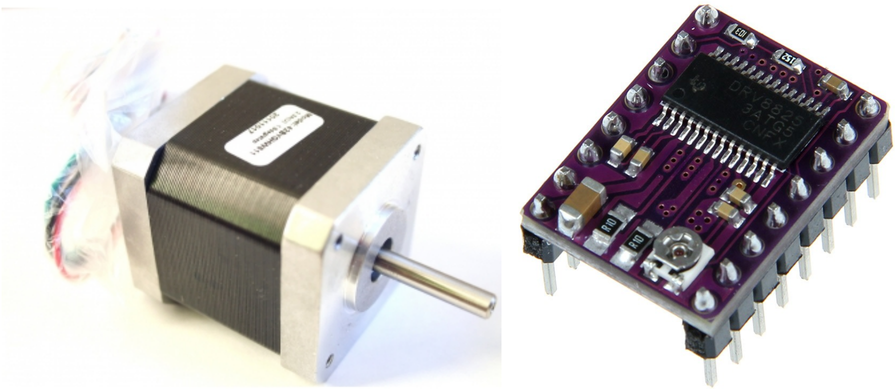
two steppers to have enough force to compensate for the gravitational forces, while the two other arms were controlled by one stepper motor each.

### NEMA 17 Stepper Motor

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotations. The stepper works by sequentially energizing it's coils, to make it rotate a step [2]. Normally, and in this project, the stepper is controlled by a stepper driver which when given a step pulse rotates the stepper motor one step, typically 1.8°. The NEMA 17 Stepper Motor, seen in Figure 5.5a, is a motor manufactured by Wantai. It has a weight of 360 g, a holding torque of 4.8 kg/cm and the shaft is measuring 5 mm in diameter. It has a current rating of 2.5 A and a voltage rating of 3.1 V.

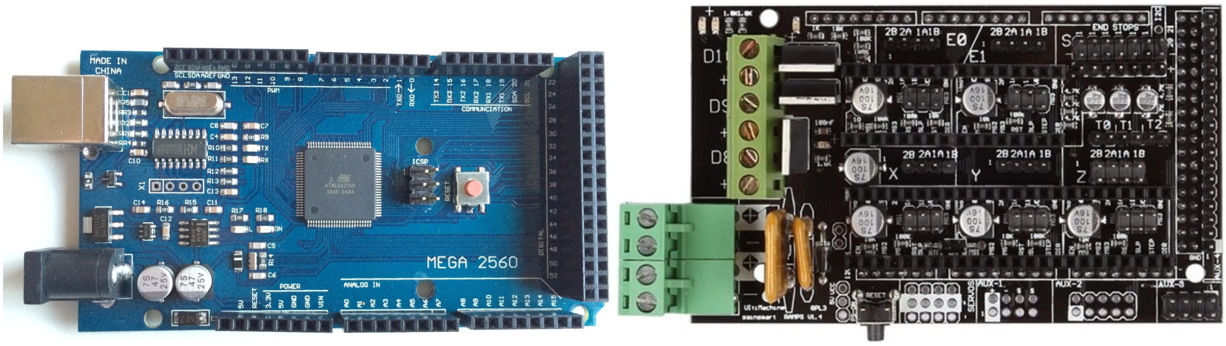
### RAMPS 2560 Electronics

The micro controller board used to control the stepper motors is the RAMPS 2560 Electronics, seen in Figure 5.6a, and is based on the ATmega 2560 micro controller, and modelled after the Arduino Mega 2560. It has 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), and a 16 MHz crystal oscillator. It has a flash memory of 256 kB and a recommended input voltage of 7-12 V.



**(a)** NEMA 17 stepper motor **(b)** Stepstick DRV8825 Driver

**Figure 5.5:** Stepper components



**(a)** Mega 2560 Microcontroller Board **(b)** RAMPS v 1.4 Shield

**Figure 5.6:** RAMPS v 1.4 basic kit

### RAMPS v1.4 Shield

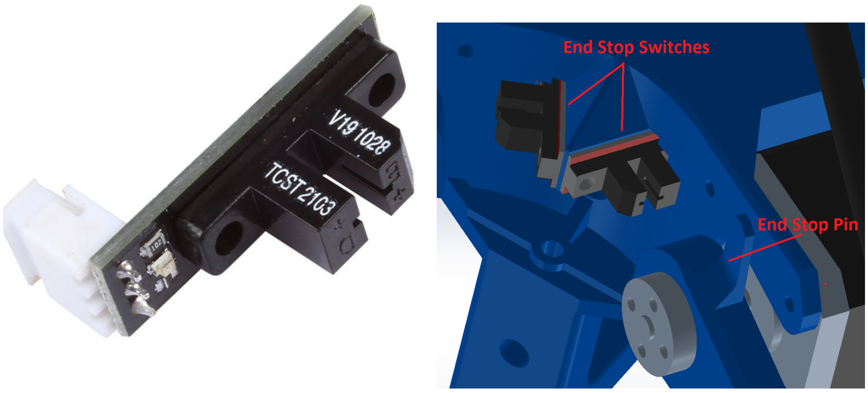
Plugged on top of the RAMPS 2560 was the RAMPS v1.4 Shield, seen in Figure 5.6b. This shield is designed for controlling a RepRap 3D-printer [14]. It has slots for five stepper motor drivers motors, six end stop switches, and a number of other input/output pins, among other things. It was chosen since the prototype use the same type of steppers as the 3D-printer.

### Stepstick DRV8825 Driver

The stepstick DRV8825, seen in Figure 5.5b, is used as interface between the micro controller board and the stepper motors. The steppers that are to be controlled have a full step size of 1.8°/step, but this resolution can be increased by the stepstick by energizing the coils differently. With the resolution pins M0-M2 on the stepsticks it is possible to increase the resolution up to 1/32 of a full step. The stepsticks are mounted on the RAMPS shield, and the interface pins have to be hardwired with jumpers that transfers 5V voltage from the shield to the pins, and for the TAC it was set to 1/2 steps, meaning 0.9°. The stepstick can be controlled by a step- and direction interface and can deliver up to 2.5A to the stepper. The board may gain high temperatures when used so it is recommended to attach heat sinks to the micro controller using thermal glue. The current level sent to the driver can be adjusted with the trimpot on the stepstick.

### Optical End Stop Switch

The TCST2300 is an optical end stop switch, as seen in Figure 5.7a, which uses light to detect if something passes between the two bars. Since the steppers has no encoder, the angular position of the stepper motor is unknown. By running an initializing function that runs the steppers until the end stop pins hit the end stop switches it is possible to keep track of the stepper's angular positions by counting steps, since the step sizes are known. This setup, which can be seen in Figure 5.7b shows how it was used in this project.



**(a)** TCST2300 **(b)** Prototype initial position setup

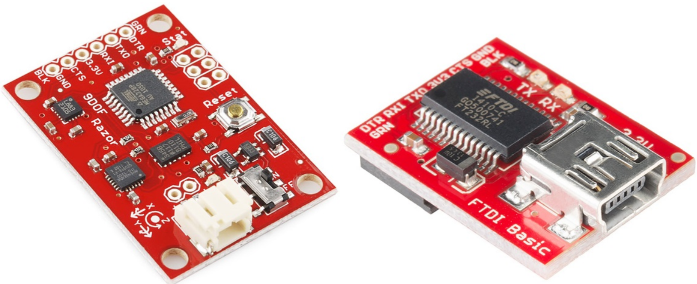
**Figure 5.7:** End stop switch

### Inertial Measurement Unit

The 9DOF Razor IMU, seen in Figure 5.8a, consist of digital triple axis gyroscope, accelerometer and a magnetometer, thus making it 9 DOF, as well as a microprocessor for processing. More information about IMUs can be found in Section 2.2.2. The IMU gyroscope is a ITG-3200, a single-chip, digital-output, triple axis microelectromechanical systems (MEMS) motion processing gyro. It has digital-output X-, Y-, and Z-Axis angular rate sensors on an integrated circuit. The operating current is 6.5mA, the standby current is 5$\mu$A and it operates with a supply voltage between 2.1V and 3.6V. The IMU accelerometer is a ADXL345, which is a triple axis MEMS accelerometer with a resolution of 13-bit at acceleration between -16$g$ and +16$g$. It operates at 25$\mu$A to 130$\mu$A at 2.5V and needs a supply voltage between 1.8V and 3.6V. The magnetometer used on the IMU is the HMC5883L which is developed by Honeywell. It has a range of -8 Gauss to +8 Gauss and a resolution of up to 5 milli-Gauss. It operates with a 2.16V to 3.6V voltage supply. The IMU microcontroller is a ATmega328. The gyroscope-, accelerometer- and magnetometer data are sent to the ATmega328, using Inter-Integrated Circuit (I2C), where it is processed before it is forwarded through the Serial Peripheral Interface (SPI) or Universal asynchronous receiver/transmitter (UART) to other devices for further use. The IMU was in this project placed inside the TAC baseplate.

### Arduino Uno

The micro controller board used to communicate with the IMU was an Arduino Uno and is a simpler micro controller board than the RAMPS 2560 Electronics described earlier. The Arduino



(a) IMU (b) FTDI Basic Breakout board

**Figure 5.8:** IMU Components

Uno has a ATmega328 microcontroller, 14 Digital I/O Pins, 6 Analog Inputs, a 32kB Flash memory, a clock speed of 16MHz and works on an input voltage of 7V to 12V.

#### IMU Communication Device

To program the Razor IMU a SparkFun FTDI Basic Breakout - 3.3V was used. The device comes with 6 female pins from the manufacturer, so to be able to connect the device to the IMU 6 male pins were soldered onto the IMUs serial port. The device was then connected to the IMU using these pins and connected to the PC using a USB-mini to USB cable. A picture showing the FTDI Basic Breakout can be found in Figure 5.8b.

### 5.3.2 Circuit Schematic and Wire List

Figure 5.9 shows a circuit schematic of the prototype configuration. Circuit names can be matched with the ones in Table 5.2 to get more info about the components. Table 5.3 is the prototype wire list that contains all the wire connections for the prototype. For the SPI to work on the Razor IMU, the RX1 pin for the UART had to be utilized as the SPI slave-select. This was done by reprogramming the IMU using the FTDI Breakout.

### 5.3.3 Handling Vibration Noise

The TAC arms are controlled by four stepper motors which generate vibrations when running. The initial plan when building the prototype was to place the IMU inside the TAC baseplate. The vibrations caused by the steppers will cause noise in the IMU measurements, thus corrupting the data. In combination with the digital filters, as described in Section 5.1, a rack to mount the TAC on was built to further attempt to suppress the vibrations.

Figure 5.10 shows the rack. The spine of the rack was created by a 0.5m long 2x1 inches thick plank. The TAC was mounted on top of this plank with rubber dampers between the TAC and plank. Further the IMU was mounted under a plank shelf 10cm under the TAC, also with damping material between the IMU and the plank. This vertical plank was bolted down to a large plate to stabilize the structure.

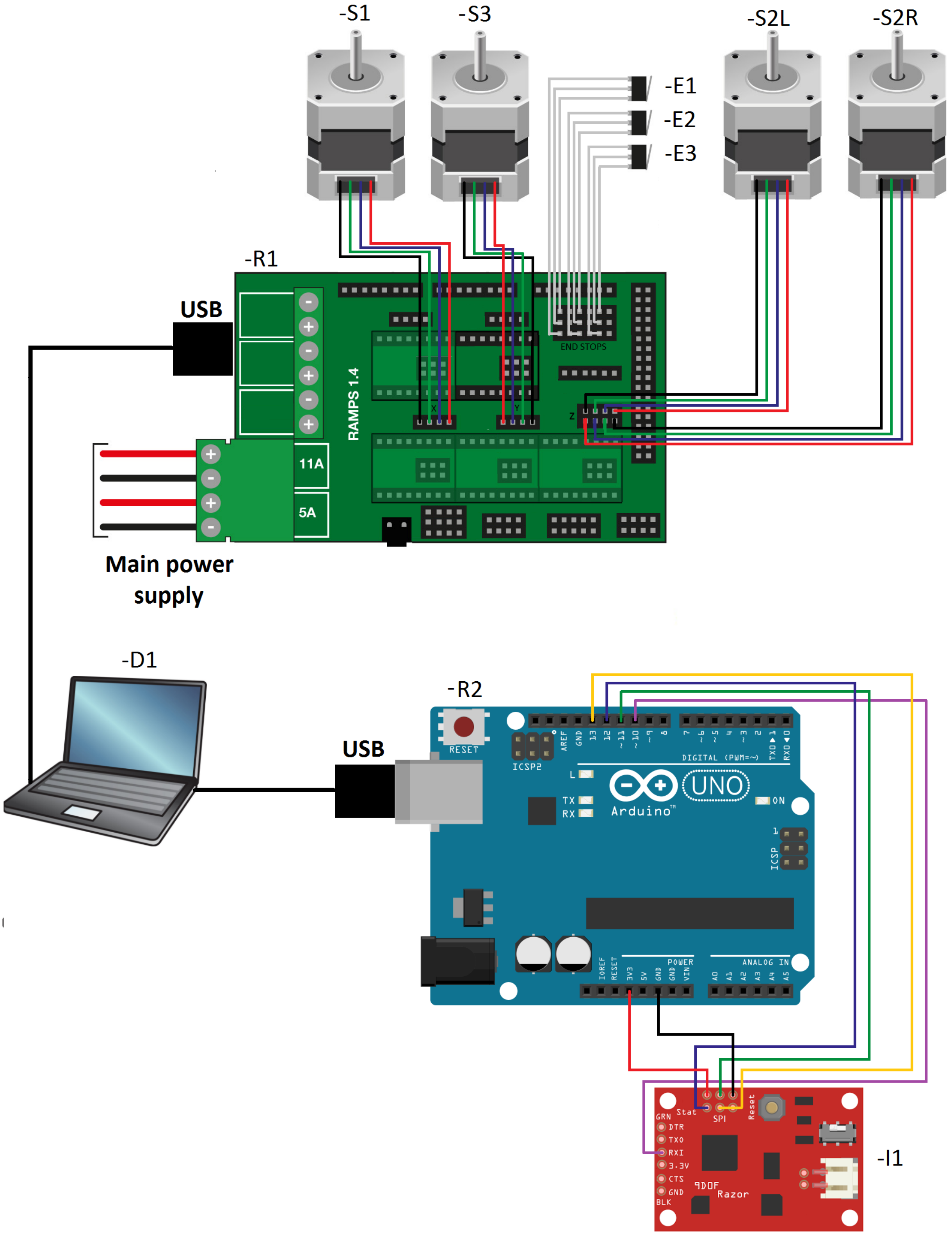


**Figure 5.9:** Prototype circuit schematic

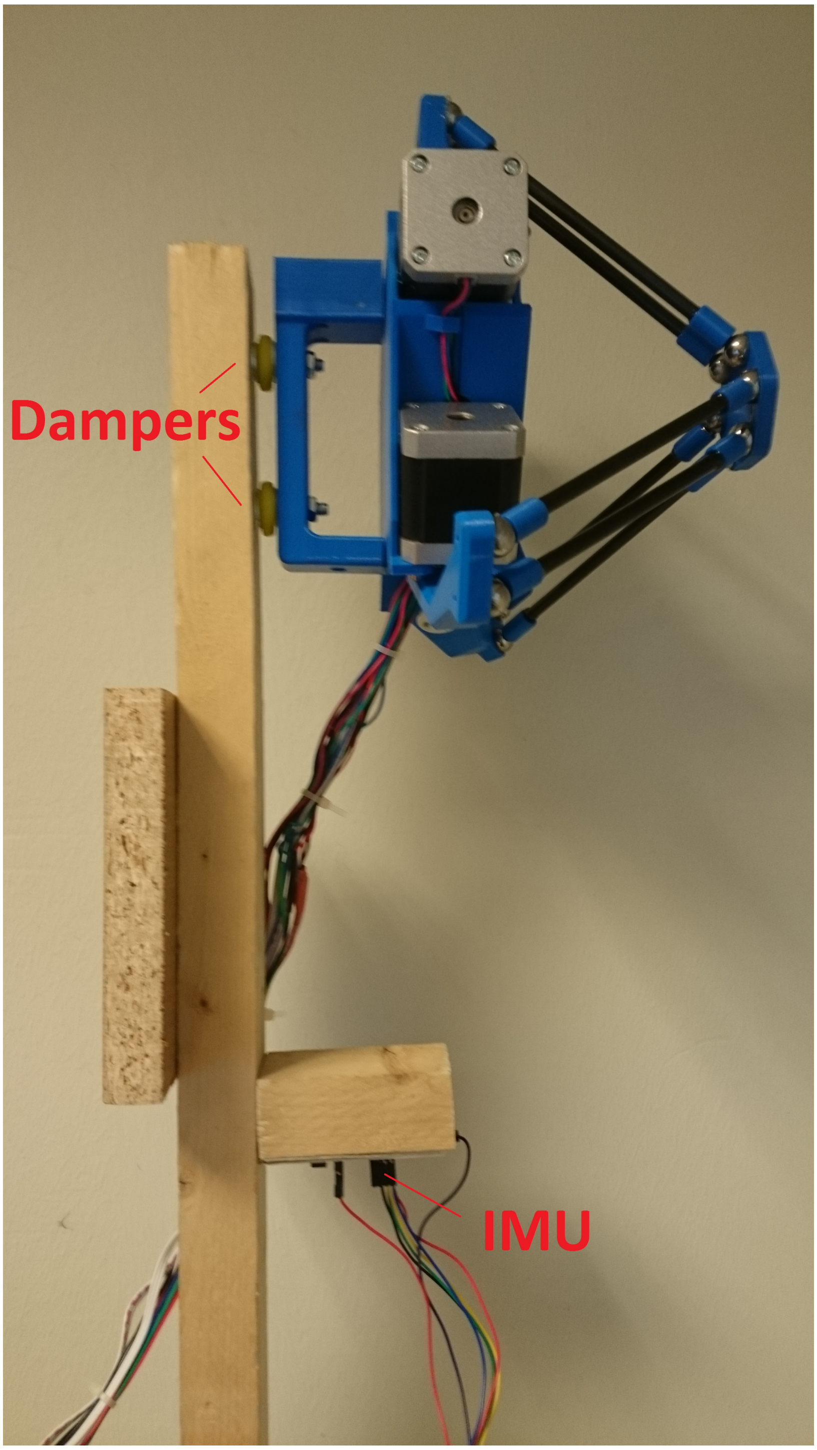


**Figure 5.10:** TAC mounted on rack, showing dampers and IMU placement

**Table 5.3:** Prototype wire list

| Intern | Colour | External |
|---|---|---|
| -R1:X2B | Black | -S1:1B |
| -R1:X2A | Green | -S1:1A |
| -R1:X1A | Blue | -S1:2A |
| -R1:X1B | Red | -S1:2B |
| -R1:Y2B | Red | -S3:2B |
| -R1:Y2A | Blue | -S3:2A |
| -R1:Y1A | Green | -S3:1A |
| -R1:Y1B | Black | -S3:1B |
| -R1:Z2B | Red | -S2R:2B |
|  | Black | -S2L:1B |
| -R1:Z2A | Blue | -S2R:2A |
|  | Green | -S2L:1A |
| -R1:Z1A | Green | -S2R:1A |
|  | Blue | -S2L:2A |
| -R1:Z1B | Black | -S2R:1B |
|  | Red | -S2L:2B |
| -R1:ENDSTOPS X-1 | Grey | -E1:S |
| -R1:ENDSTOPS X-2 | Grey | -E1:G |
| -R1:ENDSTOPS X-3 | Grey | -E1:V |
| -R1:ENDSTOPS Y-1 | Grey | -E2:S |
| -R1:ENDSTOPS Y-2 | Grey | -E2:G |
| -R1:ENDSTOPS Y-3 | Grey | -E2:V |
| -R1:ENDSTOPS Z-1 | Grey | -E3:S |
| -R1:ENDSTOPS Z-2 | Grey | -E3:G |
| -R1:ENDSTOPS Z-3 | Grey | -E3:V |
| -R1:USB | - | -D1:USB |
| -R2:USB | - | -D1:USB |
| -R2:3.3V | Red | -I1:3.3V |
| -R2:GND | Black | -I1:GND |
| -R2:12(MISO) | Blue | -I1:MISO |
| -R2:11(MOSI) | Green | -I1:MOSI |
| -R2:13(SCK) | Yellow | -I1:SCK |
| -R2:10(SS) | Purple | -I1:RX1(SS) |

## 5.4 Software

This section describes the software developed to communicate with, and control, the TAC prototype. For the system to work, the control board for the steppers, the control board for the IMU, and the IMU itself had to be programmed, as well as a MATLAB script to govern the whole system on computer.

### 5.4.1 Stepper Library

As mentioned in Section 5.3.1, the stepper motors are controlled by energizing the coils inside the stepper in sequence. The steppers are energized by sending a high signal the the motor driver mounted on the Ramps v1.4 Shield described in Section 5.3.1. One step, of a length predetermined by the resolution pins under the stepper driver, is taken when the signal is changing from low to high. The delay between step pins going from high to low is used as a method of controlling the angular speed of the steppers. The time delay for the individual stepper $\Delta t_i$ is measured in ms, and thus the stepper speed in deg/s, $\dot{\alpha}_{i,ref}$, can be found as

$$\dot{\alpha}_{i,ref} = \frac{\Delta s}{\Delta t_i}\frac{1}{1000} \quad \forall i \in \{1, 2, 3\} \tag{5.12}$$

where $\Delta s 0.9°$ is the stepsize for the prototype.

To simplify usage and organization of code and enhance code readability a library for the functions tied to the stepper motors was developed. The stepper library consists of nine different functions in addition to the class constructor. The functions are described further in Table 5.4.

**Table 5.4:** Stepper library functions

| Function Name | Type | Input | Description |
|---|---|---|---|
| enableStepper | void | bool enable | Enables stepper if input is high, disable if low |
| setInitialAngleSpeed | void | - | Sets angle speed for initialization |
| initiateStepper | bool | - | Moves stepper until endstop is detected. Returns true when detected |
| setAngleSpeed | void | float velInput | Calculates step delay from input angle speed |
| getAngleSpeed | float | - | Returns active angle speed, calculated from step delay |
| moveConstantSpeed | bool | float aref | Move with non-changing speed to input angle. Returns true when moving |
| moveVelInput | void | float velInput | Function to use in the closed loop for stepper control |
| moveInDirection | void | float input | Function that sends direction and step signal to stepper based on the size and polarity of the input |
| stepMotion | void | bool direction | Function that controls the stepper and direction pins |

### 5.4.2 Stepper Controller Code

The micro controller that is communicating with the stepper motors, R1, has the task of controlling the steppers, reading/writing from/to MATLAB. The control algorithm starts after MATLAB connects to the controller, and is described below.

**Algorithm 5.1** Algorithm for stepper controller

```
loop
    Wait for initialize command from MATLAB
    if Enable command received then
        Enable steppers
        Move steppers in the positive direction until end stop switches are detected
        Move steppers to initial position
    end if
    Wait for regulate command from MATLAB
    Enable step timers
    while Regulate mode is on do
        Send stepper angles to MATLAB
        Check for received data packet from MATLAB
        if Packet consist of control inputs then
            Change angle speed of steppers by calculating step delays
            Change direction of step motion by checking polarity of angle speed
        else if Packet consist of shutdown message then
            Turn regulate mode off
            Break
        end if
        for all steppers do
            if step timer>step delay then
                if Angle limits are not met then
                    Move a step
                end if
                Reset timer
            end if
        end for
    end while
    Disable steppers
end loop
```

### 5.4.3 IMU Controller Code

The Arduino Uno has the sole purpose of sending data from the IMU to MATLAB. The Uno is checking for new data from the IMU on the SPI port. When data arrives it is sent to MATLAB using the USB interface.

### 5.4.4 IMU Code

The code on the IMU is largely unchanged from the manufacturers side, using the Sparkfun 9DOF Razor IMU AHRS (Attitude and Heading Reference System) Firmware v1.0, which read the data from the three sensors on the board from the I2C. The modifications done for the prototype is such that it transmits acceleration in the body frame and Euler angles.

### 5.4.5 MATLAB Control and Signal Filtering Code

The two micro controllers communicate via serial protocol to the computer running a MATLAB script, which works as the brain of the system. It does all the necessary calculations in the control loop, and transmits the control inputs to the system. The procedure is described below.

**Algorithm 5.2** Algorithm for MATLAB Control and Signal Filtering

```
Require: TAC parameters, filter constants, controller gains and simulation time
Require: Choice of acceleration or attitude feedback controller
  Set up serial communication with both micro controllers
  Send initiate command to stepper controller
  Wait two seconds, approximate time stepper initiation takes
  Send regulate command to stepper controller
  Start simulation timer
  while Timer < Simulation time do
      Check for received data packet from IMU Controller
      if Data packet consist of acceleration and Euler angles then
          Process data as described in Section 5.1
      end if
      Check for received data packet from stepper controller
      if Data packet consist of stepper angles then
          Calculate new TCP position
          if Acceleration feedback controller chosen then
              Calculate Jacobian matrix
          end if
      end if
      if New IMU measurements- or New stepper angles received then
          if Acceleration feedback controller chosen then
              Calculate control input based on algorithm described in section 5.2.1.
          else if Attitude feedback controller chosen then
              Calculate control input based on algorithm described in section 5.2.2.
          end if
          Send control input to the stepper controller
      end if
  end while
  Send shutdown message to stepper controller
  Shut down the communication to both controllers
```

Chapter 6

# Experimental Results

This chapter describes the experimental results achieved when testing the TAC prototype. This will include results from signal processing, the attempt to handle vibration noise through a constructed rig, as well as the different control approaches.

## 6.1 Signal Processing

This section will present the results from the filtering and integration method described in section 5.1. The magnitude spectra are found using the technique described in section 2.3.1, and analyzed to dimension the filters.

Figure 6.1 compares $\mathbf{\Theta}_x$ with $\mathbf{\Theta}_y$. As can be seen, $\mathbf{\Theta}_y$ is a lot smoother than its unfiltered counterpart $\mathbf{\Theta}_x$, at the cost of some latency. Figure 6.2 shows the magnitude spectra of $\mathbf{\Theta}_x$, $|\mathbf{\Theta}(f)_N|$, along with the cutoff frequency for the LPF.

Figure 6.3 compares $\mathbf{a}_x^t$ with $\mathbf{a}_y^t$. Similar to the attitude measurements, it is apparent that the filtered signal is smoother than the unfiltered, at the cost of some delay. The removal of a bias is apparent in the $y$-axis, but this may occur in any of the axes. Figure 6.4 shows the magnitude spectra of $\mathbf{a}_x^t$, $|\mathbf{A}(f)_N|$, along with the cutoff frequencies for the LPF and HPF.

Figure 6.5 compares $\mathbf{v}_x^t$ with $\mathbf{v}_y^t$. As $\mathbf{v}_x^t$ is a smooth signal, low pass filtering is not necessary, but because the values tend to drift when integrating a signal, a HPF was used. This is noticeable as the filtered signal stay around zero, as the unfiltered signal drifts. Figure 6.6 shows the magnitude spectra of $\mathbf{v}_x^t$, $|\mathbf{V}(f)_N|$, along with the cutoff frequency for the HPF.

Figure 6.5 compares $\mathbf{p}_x^t$ with $\mathbf{p}_y^t$. Similar to the velocities, the smoothness means no LPF is required. $\mathbf{p}_x^t$ may drift, which is why a HPF was added, although no obvious bias can be seen in Figure 6.5. The resulting $\mathbf{p}_y^t$ is the only estimate of the positional deviation of the TAC, and the movements done were quite similar. Figure 6.8 shows the magnitude spectra of $\mathbf{p}_x^t$, $|\mathbf{P}(f)_N|$, along with the cutoff frequency for the HPF.

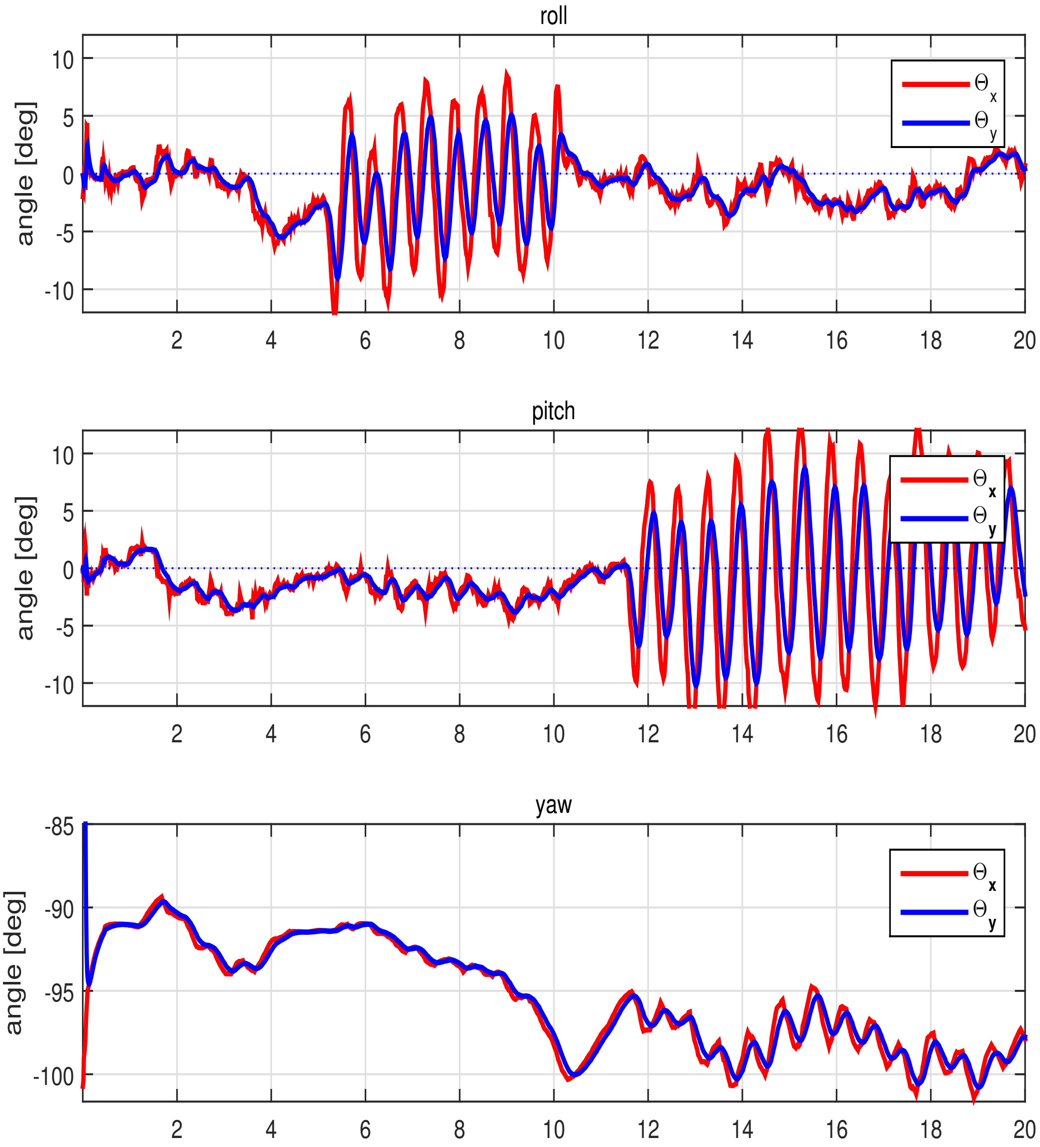


**Figure 6.1:** Comparison of unfiltered- and low passed filtered Euler angles

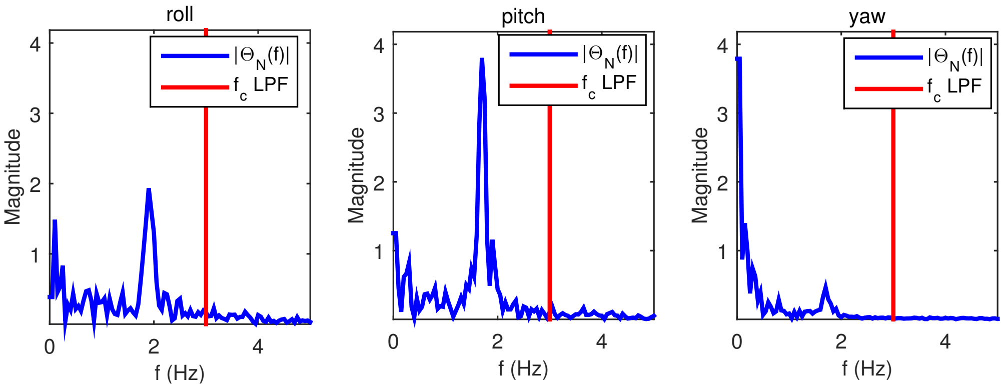


**Figure 6.2:** Unfiltered Euler angles magnitude spectra

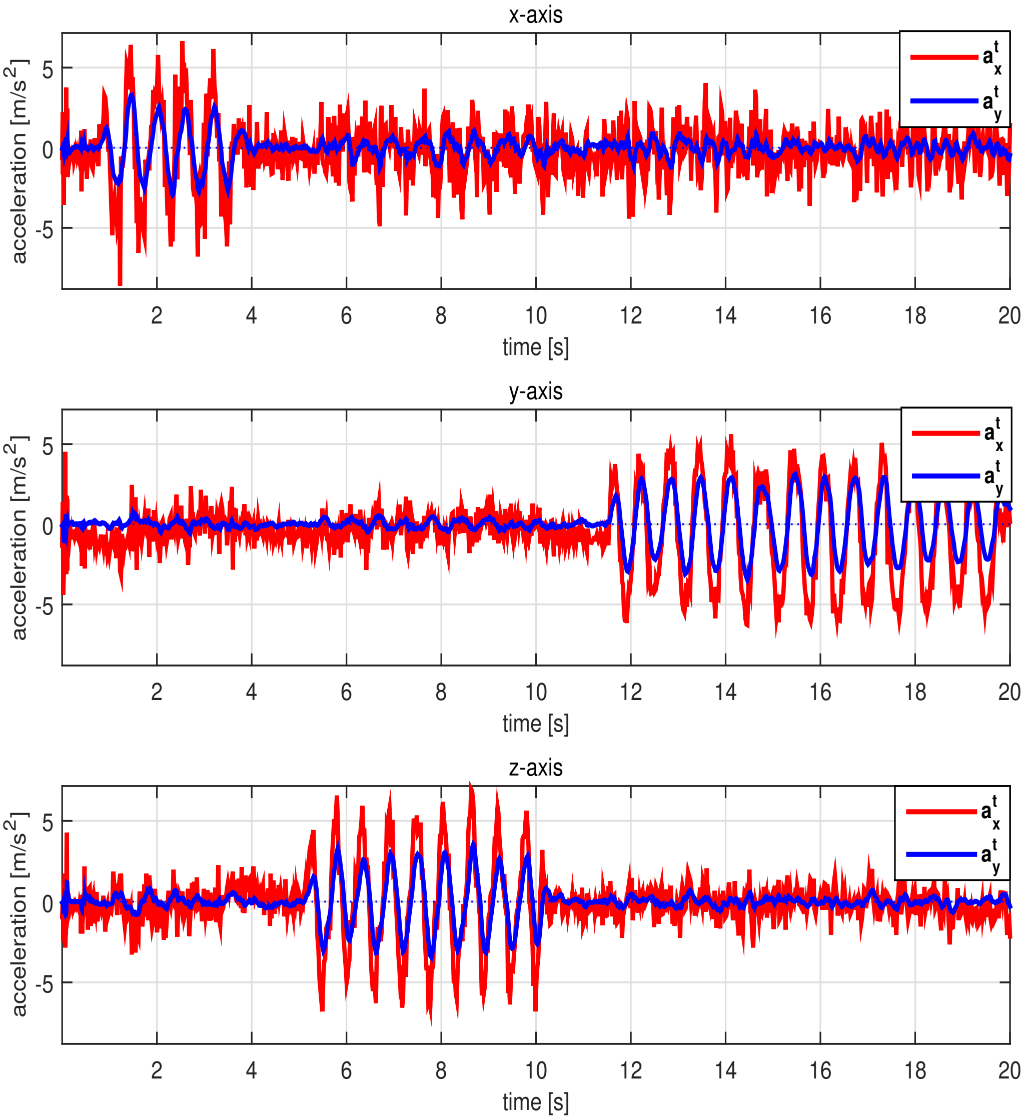


**Figure 6.3:** Comparison of unfiltered- and low- and high passed filtered accelerations

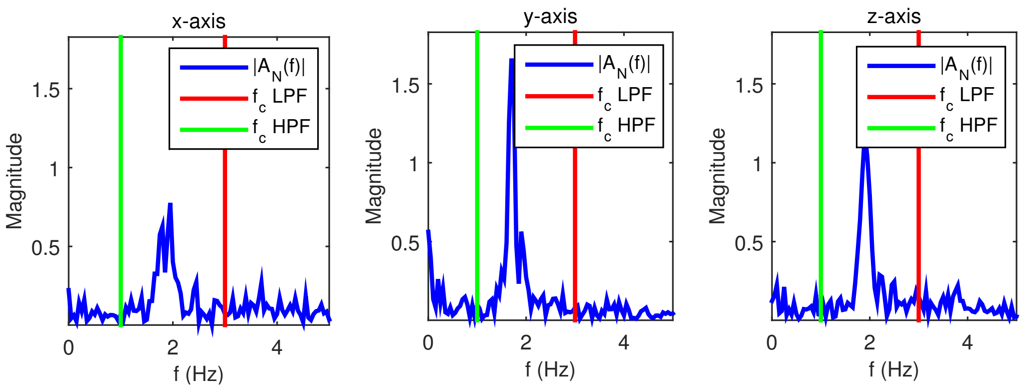


**Figure 6.4:** Unfiltered acceleration magnitude spectra

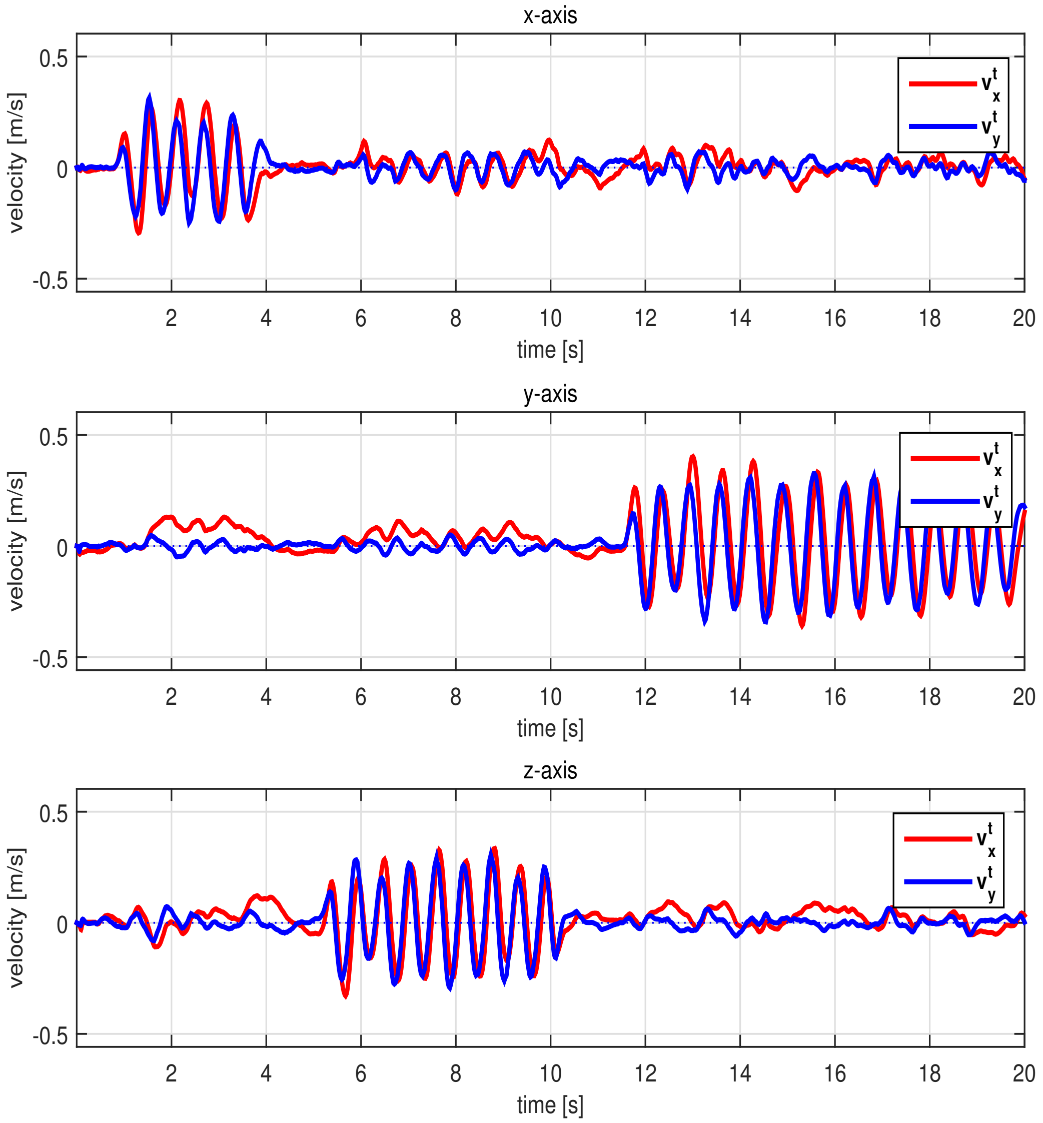


**Figure 6.5:** Comparison of unfiltered- and high passed filtered velocities

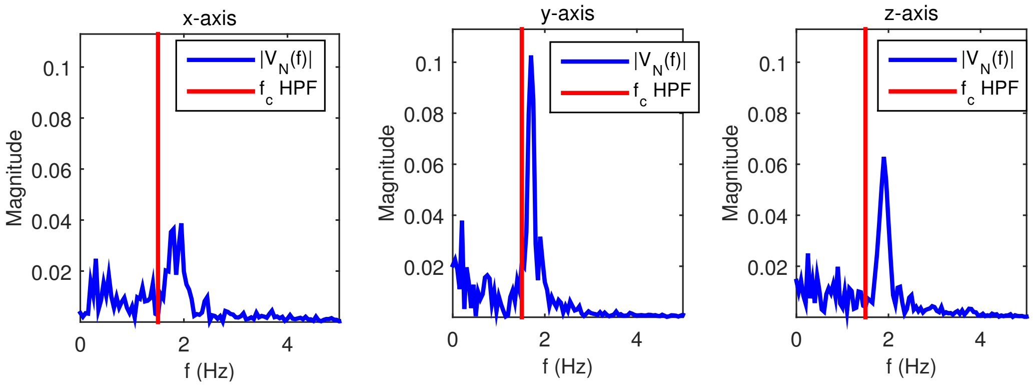


**Figure 6.6:** Unfiltered velocities magnitude spectra

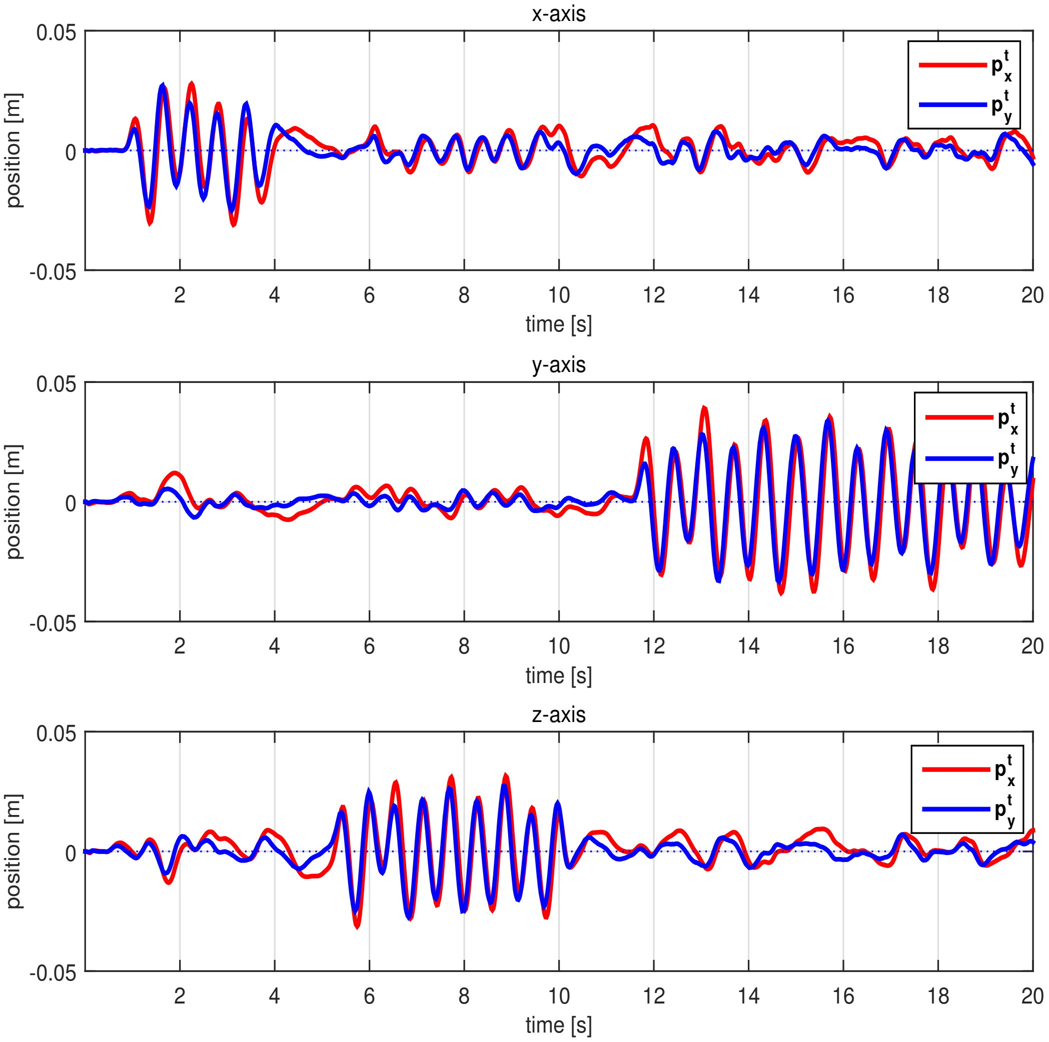


**Figure 6.7:** Comparison of unfiltered- and high passed filtered position

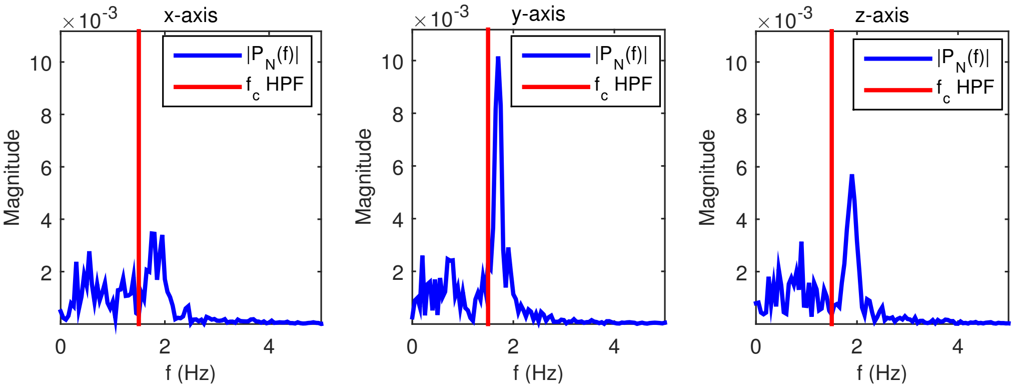


**Figure 6.8:** Unfiltered position magnitude spectra

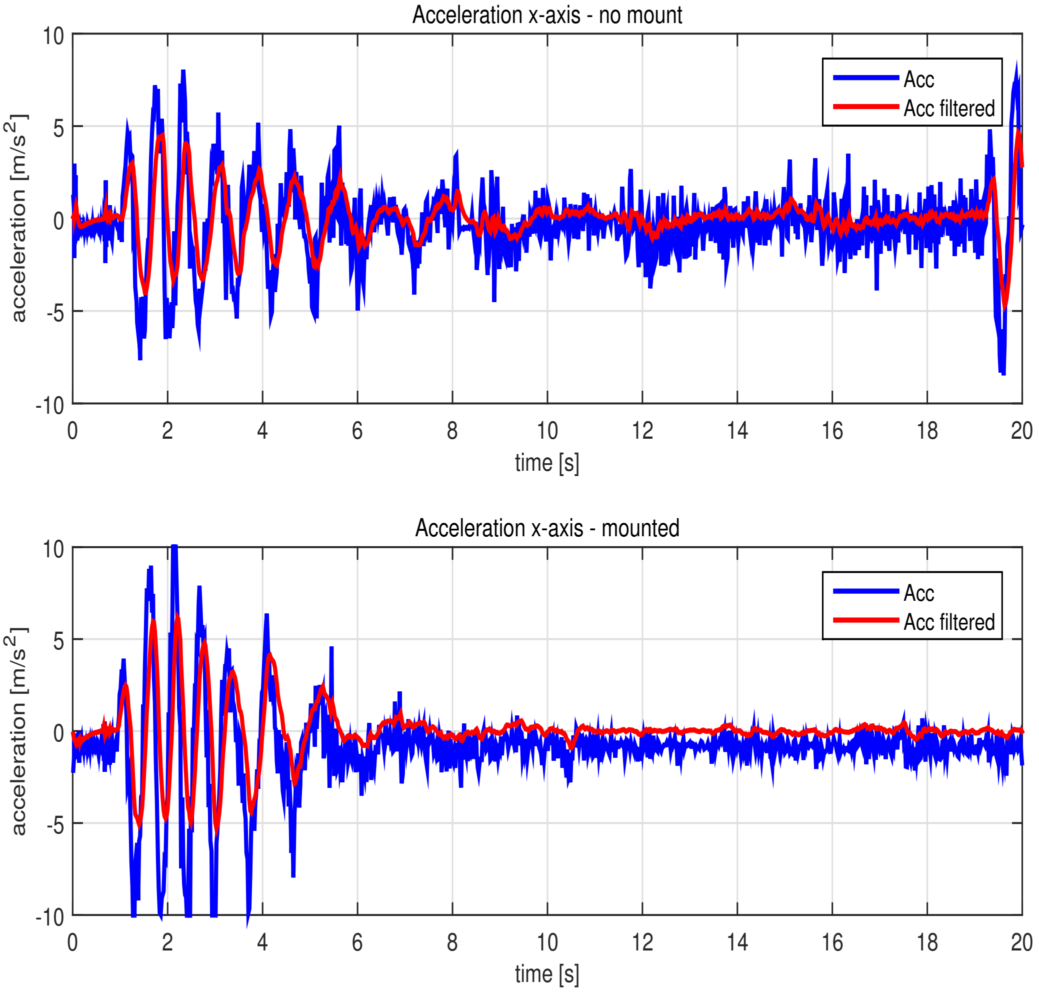


**Figure 6.9:** Comparison of acceleration measurement when TAC is handheld and mounted

## 6.2 Handling Vibration Noise

Figure 6.9 compares $\mathbf{a}_x^t$ from the mounted and unmounted TAC when the TAC was moved in a sweeping motion in the x-direction with about the same amplitude and decreasing frequency. The amplitude of the noise in the acceleration measurements is somewhat reduced when the TAC is mounted on the rack compared to when it was not mounted. It is still difficult to detect slower movements. Both the mounted and unmounted TAC failed to detect movements with a period bigger than about 1.5 seconds. It is worth mentioning that is difficult to create the exact same experiment two times, especially when one TAC is handheld and the other is mounted.

## 6.3 Control Approaches

This section will present the results from the control approaches described in Section 5.2.

### 6.3.1 Closed Loop TAC Control with Acceleration Feedback

The closed loop TAC controller with acceleration feedback was tested by moving the TAC around at different speed and angles. Because of the bandwidth chosen, the system is able to detect fast movements, but slow movements gets suppressed with the noise by the filters. The TAC did some compensation for fast movements, both linear and angular, but did not try to compensate for slow movements. Since the IMU is placed inside the TAC baseplate, the two frames $\{b\}$ and $\{t\}$ are equal. And since the desired position for the TAC should be constant, $\boldsymbol{p}_d^n = \boldsymbol{p}_d^t$. The TCP position in $\{n\}$-frame can be estimated by using to returned TCP position, $\boldsymbol{p}_c^t$, and the estimated displacement, $\boldsymbol{p}_y^t$, to

$$\boldsymbol{p}_c^n = \boldsymbol{p}_c^t + \boldsymbol{p}_y^t, \tag{6.1}$$

which will be used for plotting. Figure 6.10 compares $\boldsymbol{p}_c^n$ to the desired position $\boldsymbol{p}_d^n$ and the estimated uncompensated position $\boldsymbol{p}_s^n$, which is the position the TCP would have if the steppers were locked to $\boldsymbol{\alpha}_d$, and is found as

$$\boldsymbol{p}_s^n = \boldsymbol{p}_d^t + \boldsymbol{p}_y^t. \tag{6.2}$$

In this experiment, the TAC was first moved up and down in the $x$-axis, then side to side in the $z$-axis, and finally side to side in the $y$-axis, as can be seen by the oscillations in the figure.

It can be seen that $\boldsymbol{p}_c^n$ stays close to the reference signal, and during the oscillatory movements, it counteracts the movements for a while, before latencies cause it to be in sync with them. Also there can be seen some drift in the $x$-axis. Both $\boldsymbol{p}_c^n$ and $\boldsymbol{p}_s^n$ are based on the estimate $\boldsymbol{p}_y^t$, which means that the plots may not be entirely accurate.

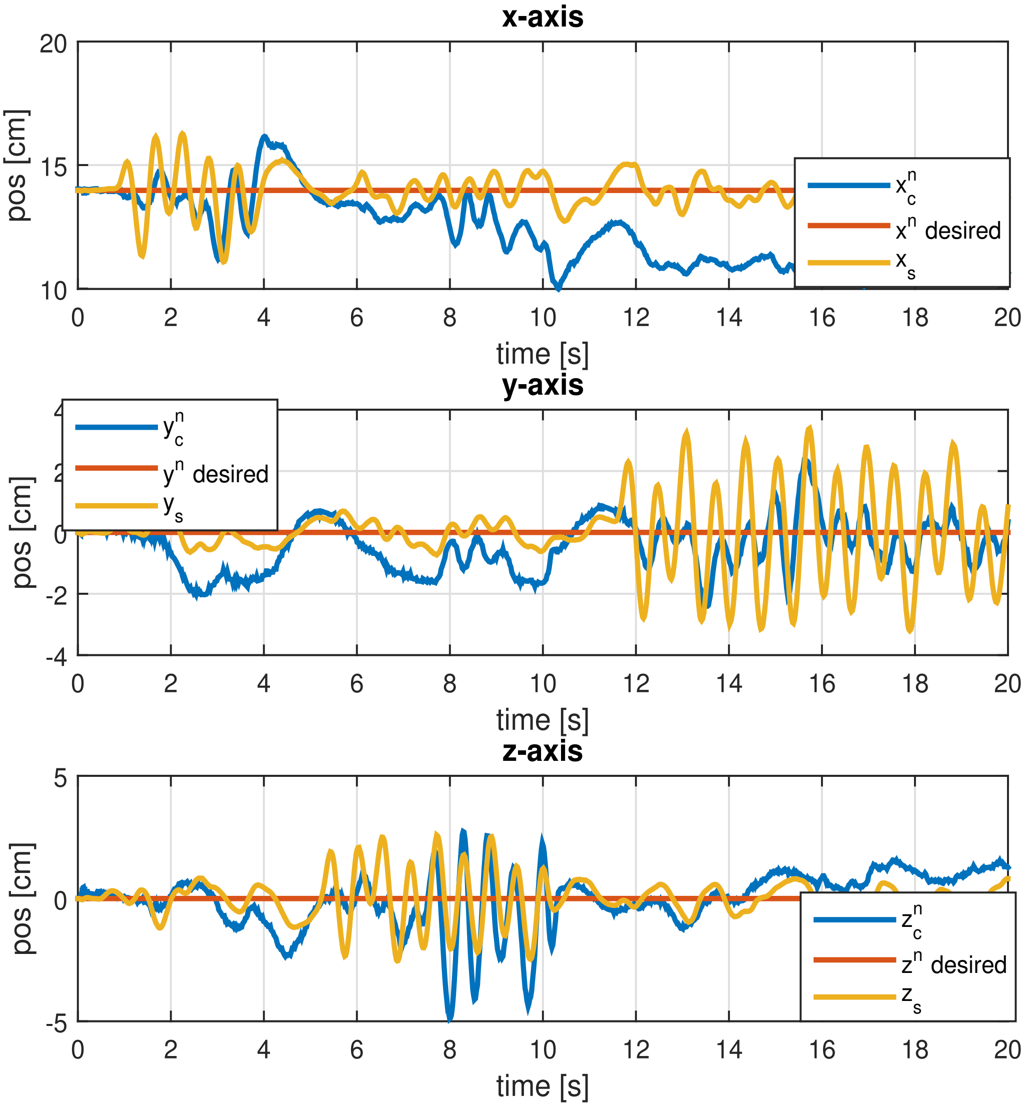


**Figure 6.10:** Comparison of estimated position and reference using acceleration feedback

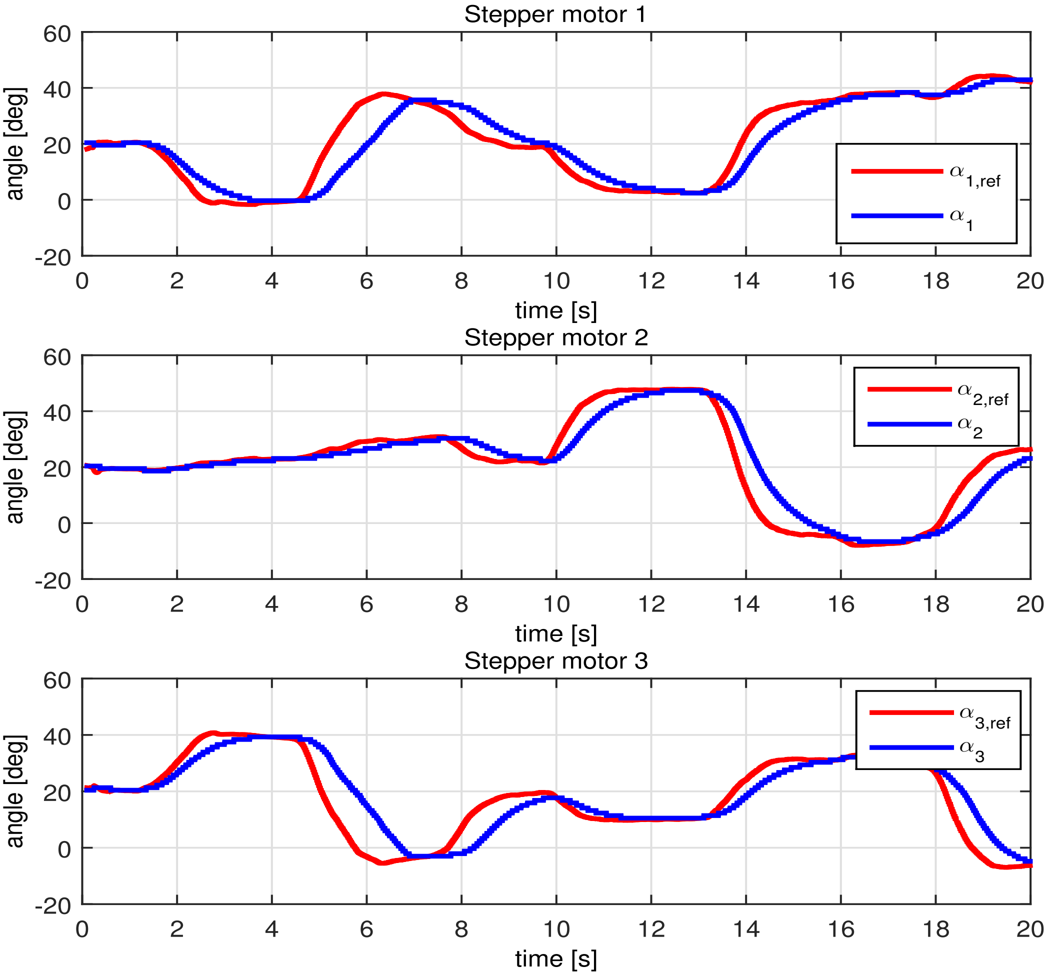


**Figure 6.11:** Stepper angle response for the attitude controller

### 6.3.2 Closed Loop TAC Control with Attitude Feedback

The closed loop TAC controller with attitude feedback was tested by tilting the TAC at different angles. When tilting the TAC with slow rotations the TCP always sought to retain the same position in the $\{n\}$-frame. When tilting the TAC with faster rotations TCP started to oscillate and became unstable.

Figure 6.11 compares the stepper angles, $\boldsymbol{\alpha}$, to the reference stepper angles, $\boldsymbol{\alpha}_{ref}$, when the TAC was tilted with slow movements. It can be observed that $\boldsymbol{\alpha}$ are following $\boldsymbol{\alpha}_{ref}$ well, although they are delayed. By increasing the controller gain the system got unstable, and when decreasing it the system became to slow for the steppers to follow the reference.

# Chapter 7
# Discussion

The work described in this thesis was conducted over the period of six months, and touches on a several different topics and work methods, with differing results. Both the TAC simulator and prototype could be improved in different ways, although the importance of the improvements might be negligible for the final result. The TAC, as described in this thesis as the merging of a Delta robot and an IMU, is a non-linear structure with non-linear sensory data. As such, the fragments required to achieve the objective were many, and not all so obvious. This chapter will discuss the solutions found and ideas for further work.

## 7.1 Modeling and Simulations

The designed simulator included a university approved vessel simulator of a supply type ship with full equations of motion for the ship, where the size of the crane was chosen appropriately to fit with the size of the ship. The vessel simulator could be perturbed with a wave generator, with the most significant parameter assigned as of significant wave height. The DOFs from the ship are output directly as IMU measurements used in the control loop of the TAC, which is not particularly realistic for a real situation, as there would be measurement noise. Thus, the signal would have to be filtered, just like for the prototype. Another issue with the output DOFs was the drift, causing the ship to move it's position, and made it harder for the TAC to compensate. This could possibly have been corrected with a high-pass filter of the DOFs, or by adding Dynamic Positioning (DP) system to the vessel simulator.

The crane simulator, on the other hand, was highly simplified. The crane's body had no real dynamics, meaning no mass, effects of gravity or other forces of nature, moments of inertia or motor dynamics governing it's movements. It was moved merely by changing joint angle and angle speed, which was required to calculate the setpoint for the TAC. Thus, simplified crane dynamics are enough to show how to use the crane's movements to calculate the necessary counteractions to keep the TCP in a compensating fashion. With a more complicated crane model, a separate control loop for this could be implemented to control the trajectory of the TAC, with patterns such as loading and offloading of cargo, while the TCP compensates for the disturbances.

Also worth to mention is the TAC simulator itself. It included a first-order function for the DC-motors, and the Jacobian matrix to transfer the angular speed from the DC-motors to the position of the TCP. This is a simplified simulator, but it had some motor dynamics, and therefore something to regulate, albeit easy to yield convergence.

Finding the setpoint to keep the TCP in the right place can be an important part of the control algorithm, but more important is the situation of the cargo it holds. The dampening of the cargo's movement might not be compatible with keeping the TCP in the same spot, but controlling the position will no doubt be a part of it. The dynamics of suspended cargo would need modelling, which could influence the crane itself.

With a more advanced system model it would require better tuning of the controller, as the response would be less controllable. This could possibly be cause to implement a more advanced system controller, such as Model Predictive Control (MPC), where the whole crane is considered for optimal control.

## 7.2 Prototype

### 7.2.1 Construction

Building the TAC prototype has been an interesting and challenging affair, moving the focus from merely theoretical applications to more practical efforts. The prototype construction was done using 3D-printed parts, combined with electrical components and micro controllers, as described in detail in Chapter 5.

The 3D-printed parts were made out of ABS plastic, printed at NTNU, and a problem with this was the strength of the end stop pins, which were used to ascertain the angles of the arms. These pins would break very easily, which happened a bit too often during the development of the software for the prototype. A hard coded limiter for the stepper motors was programmed which reduced the breakage frequency. Different material for these pins could have been used to make them stronger.

A smaller issue with the TAC prototype was the fact that the steppers' movements caused disturbances in the IMU readings when the IMU is placed inside the baseplate of the IMU. A rectifying maneuver was attempted by building a rig for the TAC, as described in section 5.3.3. However this only improved the sensor data slightly, as seen in section 6.2. As described in Chapter 3, the IMU on a ship is normally placed inside the body of the ship, such that there might be several other sources of noise than the crane motors to be filtered. The rig was only tested for linear motions, and algorithms for including the difference in location of the TAC and IMU was not implemented, but would be interesting for further work.

### 7.2.2 System Control

The Mega 2560 that was used to control the stepper motors was at first used for all code and calculations, as well as communicating directly with the IMU. The idea was to use it as the only controller for the TAC, such that it did not need be connected to anything but a power source, but some issues from this configuration caused the inclusion of a computer running MATLAB, and a separate controller for the IMU.

The first issue pertain to computational power required in the algorithm. They include a lot of floating point math, matrix multiplications and matrix inversions, among other things, which are quite strenuous for the micro controller. These types of operation are easily done in MATLAB with i7-processor as used in the experiment, but on the micro controller it is a lot more strenuous. As such, it made sense to do the calculations in MATLAB, not only because of the computational costs, but also since it made the programming easier. For a more compact system, a micro controller specifically designed to do more advanced mathematical calculations could be utilized to circumvent this issue.

The second and most important issue has to do with the RAMPS shield and the micro controller, described in section 5.3.1. The IMU is set up to communicate with either serial or SPI, but when connected to the same micro controller that controlled the steppers, the readings from the IMU were corrupted because of Electromagnetic Compability (EMC) issues, originating from the steppers. It was discovered that sending the sensor data over USB wire directly using the serial protocol to the micro controller was possible without corrupting the data while the steppers were running. To get the sensory data, a second micro controller, an Arduino Uno, was used to receive the SPI transfers, and then both micro controllers communicated with a computer running a MATLAB script, thus splitting the control and sensor circuits. For future work, a new type of shield could have been constructed, with separate control and sensor circuits such that one micro controller could do all the work, simplifying the hardware setup. Also SPI is known to be faster than serial, so if the only communication was between the IMU and a micro controller through SPI, instead of three nodes using serial as it is now, the response time might be quicker.

### 7.2.3 Results

The prototype built as a handheld TAC, which was enough for the authors to show how it is possible to close the control loop using the IMU. However, it could been interesting to build a full crane as described in Chapter 3 to test the algorithms described in section 4.2. With a full crane structure, realistic wave patterns might be easier to generate, to prove even further that the TAC can be used as the ship's crane head. This could perhaps also been accomplished by combining the TAC prototype with the TCP patterns calculated in the simulator, and see if the TAC prototype is able to move in the same fashion as the simulated TAC, in a Hardware-in-the-loop (HIL) test.

The results of the experiments can be hard to measure, as the only sensor is the IMU. The only feedback on the position of the TAC is either integrated and filtered acceleration, and since LPF's are used there will be delays in the filtered signals. In section 6, the validity of the experiment was checked using the integrated position deviation of the TAC, which is a mere estimate of the actual positional deviation. Other sensor systems in combination with the IMU, for example GNSS, would be a good idea for further testing. With some feedback on the integrated velocity or position a better estimate could be attained, which would lead to better motion compensation. Including a Kalman filter in the estimation process could also be a good idea to improve the estimates.

Using acceleration feedback, the results were good when the movements were inside of a certain frequency range. If the TAC was moved too slowly, the usable measurements were filtered away with the noise. With the Euler angle feedback, slow rotations were successfully compensated for. However, the controller had a hard time keeping up with the movements at higher velocities, as the latency increased and the system could become unstable. This might be fixed by including a derivative part in the regulator.

These two different feedback controllers were not combined for the prototype. The algorithm for the full crane described in section 4.2 did include both rotational and positional velocity and position, but not acceleration feedback. Thus, a controller might be designed for the prototype that combines the two controllers, to possibly get the best of both acceleration and attitude feedback control rolled into one.

Chapter 8

# Conclusion

This thesis was written with the objective to prove that a Delta robot can be used for motion compensation. The equations and algorithms needed to do this have not been a straightforward task. Because of the TAC's highly nonlinear nature, many different topics had to be researched to end up with positive results. The functions and connections detailed in Chapters 2 and 3 have been key to attain the goals set. Knowing the geometry and kinematics of the TAC have been tremendously important for both the simulator and the prototype, and the filtering techniques acquired have been indispensable to get the prototype to compensate.

One of the main products of this thesis is the work in the simulator, described in Chapter 4, which resulted in in a conference paper at the 10th IFAC Conference on CAMS. The simulator, although simplified, shows how the TAC can be used for motion compensation on a ship crane. The functions derived and the control algorithm designed proved that the set points can be calculated correctly based on the IMU data. By improving the model of the crane, one can expect a harder system to control. At the same time it is reasonable to believe that the control algorithms can be useful for motion compensation, or at least as a starting point for further development.

The other main product of this thesis was the TAC prototype, described in Chapter 5. The prototype is a scaled down model which uses an IMU to measure accelerations and attitude. The first challenge was to filter the measurements properly to be able to use them for motion compensation, and this was achieved with acceptable results, considering that the only available sensor is the IMU. To compensate for the measurement noise in the low cost IMU, the filters was tuned with a narrow bandwidth, which meant that the TAC had to be moved with a certain frequency for the motions to be interpreted correctly.

Two controllers were implemented for the TAC, one with acceleration feedback and one with attitude feedback, and both worked in some way. The controller with acceleration feedback worked acceptable moving within the accepted bandwidth, and with better sensory data it should be possible to increase this bandwidth. Controlling the TAC based on attitude feedback was achieved for slow movements, and it is reasonable that with a better controller it could be achieved for fast movements as well.

This thesis has proved that the implementation of a TAC is possible both by simulations and prototyping. In the real world a more accurate IMU would be used in combination with GNSS, resulting in better measurements, and the possibilities for better filters like Kalman filter, such that the integrated estimates do not drift so much. With a more accurate knowledge about the position and attitude of the TAC, it would certainly increase the TAC's possibility to offer good motion compensation.

# Bibliography

[1] Andrioaia, D., Pascu, M., Mihaila, L., and Obrea, C. F. (2012). Determining the workspace in case of the robots with parallel structure delta 3dof. In *Annals & Proceedings of DAAAM International 2012*, volume 23. DAAAM International.

[2] CircuitSpecialists (2016). Stepper motors. https://www.circuitspecialists.com/stepper-motor, accessed 04.06.16.

[3] Clavel, R. (1988). Delta, a fast robot with parallel geometry. In *1988 18th International Symposium on Industrial Robots (ISIR) in Lausanne, Switzerland*, pages 91–100. Berlin: Springer-Verlag.

[4] Codourey, A. (1988). Dynamic modeling of parallel robots for computed-torque control implementation. *The International Journal of Robotics Research*, 17:1325–1336.

[5] Coope, I. D. (2000). Reliable computation of the points of intersection of $n$ spheres in $\mathbb{R}^n$. *ANZIAM Journal*, pages 461–477.

[6] Egeland, O. and Gravdahl, J. T. (2002). *Modeling and simulation for automatic control*, volume 76. Marine Cybernetics Trondheim, Norway.

[7] Fang, Y., Wang, P., Sun, N., and Zhang, Y. (2014). Dynamics analysis and nonlinear control of an offshore boom crane. *IEEE Transactions on Industrial Electronics*, 61(1):414–427.

[8] Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley.

[9] Johansen, T. A., Fossen, T. I., Sagatun, S. I., and Nielsen, F. G. (2003). Wave synchronizing crane control during water entry in offshore moonpool operations - experimental results. *IEEE Journal of Oceanic Engineering*, 28(4):720–728.

[10] Küchler, S., Mahl, T., Neupert, J., Schneider, K., and Sawodny, O. (2011). Active control for an offshore crane using prediction of the vessel's motion. *IEEE/ASME Transactions on Mechatronics*, 16(2):297–309.

[11] Laribi, M. A., Romdhane, L., and Zeghloul, S. (2008). Advanced Synthesis of the DELTA Parallel Robot for a Specified Workspace. *Parallel Manipulators Towards New Applications*, pages 207–210.

[12] Messineo, S. and Serrano, A. (2009). Offshore crane control based on adaptive external models. *Automatica*, 45(11):2546 – 2556.

[13] Morrison, M. (1987). Inertial measurement unit. US Patent 4,711,125.

[14] RepRapWorld (2016). Ramps - fully assembled v1.4. https://reprapworld.com/?products_detailsproducts_id=50.V1Ka_PmLTmE, accessed 04.06.16.

[15] Selesnick, I. W. and Burrus, C. S. (1998). Generalized digital butterworth filter design. *Signal Processing, IEEE Transactions on*, 46(6):1688–1694.

[16] Shenoi, B. A. (2005). *Introduction to digital signal processing and filter design*. John Wiley & Sons.

[17] SNAME (1950). The society of naval architects and marine engineers. nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin No. 1–5*.

[18] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*. John Wiley & Sons.

[19] Stan, S.-D., Manic, M., Szep, C., and Balan, R. (2011). Performance analysis of 3 dof delta parallel robot. In *2011 4th International Conference on Human System Interactions (HSI)*, pages 215–220. IEEE.

[20] Vraa, N. K. (2014). Utviklet verdens mest stabile kran i kristiansand. http://www.tu.no/industri/2014/04/03/utviklet-verdens-mest-stabile-kran-i-kristiansand, accessed 01.12.15.

[21] Williams, R. L. (2015). The delta parallel robot: Kinematics solutions. *Mechanical Engineering, Ohio University*.

# Appendix A
# CAMS16 conference paper

# Three-axis Motion Compensated Crane Head Control [⋆]

**Vegard Wie Henriksen** [*] **Audun Gerhardsen Røine** [*]
**Espen Skjong** [*,**,⋆⋆] **Tor Arne Johansen** [*]

[*] *Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.*
[**] *Ulstein Power & Control AS, Norway.*

**Abstract:** Offshore operations can be harsh and demanding and set personnel and equipment at risk. Ships will be exposed to the elemental forces of wind, waves and current, which will influence offshore crane operations considerably. This paper addresses the use of a crane head, constructed as a Delta parallel robot, to compensate for the motions of the ship in three axes. This type of robot has a rigid and accurate structure, but because of its highly nonlinear nature, advanced control algorithms must be derived. This paper includes both forward and inverse kinematics for the robot, as well as velocity kinematics and workspace analysis. The kinematics of a full crane system, with the robot as its head, has been modelled, and a simulator which includes a model of a supply vessel is created. The disturbances perturbed on the system from the elements are translated and rotated to the crane head frame of reference for use in the compensation procedure. A PID-regulator is used to control the crane head, and simulations are conducted to verify that the crane head is able to compensate for the motions created by waves.

*Keywords:* Motion compensation, Nonlinear control, Robotics, Ship dynamics, Cranes

## 1. INTRODUCTION

Offshore crane operations in harsh environments are challenging and put crew and equipment at risk. Heave compensated crane systems in marine vessels have been extensively used to cultivate easier and safer offshore operations. Examples of such operations are surface crane operations for installing equipment on the seafloor, launching and retrieving systems (LARS) and delivering supplies from vessels to platforms. Motion compensation in such systems are generally limited to one axis, i.e the vertical heave motion, (Fang et al., 2014; Kchler et al., 2011; Messineo and Serrano, 2009; Johansen et al., 2003).

This paper considers the development of a crane head designed for motion compensation in all three axes. This Three Axis Compensator (TAC) is a Delta type parallel robot (Clavel, 1988). A parallel robot consists of two or more closed kinematic chains linking the base to the end effector, whereas a serial robot arm consists of just one kinematic chain (Spong et al., 2005). The advantages of a parallel structure is its high rigidity and accuracy, making it very attractive for crane operations, whereas the disadvantages are narrower workspace and more difficult control than its serial counterpart (Laribi et al., 2008). The Delta robot consists of three kinematic chains connected on either end at a top- and bottom plate, and these plates stay in parallel with each other (Codourey, 1988). It is most commonly used for precise and stationary actions such as item picking or 3D printing (Williams, 2015), but in this paper it will be seen that it can also be used for motion compensation of crane operations on ships, which is a novel application.

It will be discussed how to use the TAC to compensate for the motion of a load hanging from a crane on a ship at sea. First a mathematical model of the crane head will be provided. Section 2 deduces the crane head geometry which is used in Sections 3, 4 and 5 to find the inverse, forward and velocity kinematics of the TAC, respectively. The reachable limits of the TAC, the workspace limits of the TAC, is explored in Section 6. The kinematics of the full crane system, including how the measurements from the Inertial Measurement Unit (IMU) are related to the states of the system is detailed in Section 7, whereas Section 8 will tie together all the different parts required to control the TAC. The simulation setup and results are presented in Sections 9 and 10.

## 2. TAC GEOMETRY

To fully understand how the TAC can be used for motion compensation, it is imperative that the TAC's configuration can be explained and designed precisely. Fig. 1 shows a geometrical representation of the TAC with the different parameters further explained in Table 1. The Tool Center Point (TCP) is where the load is suspended, and its position is denoted $\mathbf{p}_c$.

[⋆] This work has been carried out at the Centre for Autonomous Marine Operations and Systems (NTNU AMOS). The Norwegian Research Council is acknowledged as the main sponsor of NTNU AMOS. This work was supported by Ulstein Power & Control AS and the Research Council of Norway, Project number 241205.

[⋆⋆] Corresponding author: espen.skjong@ulstein.com

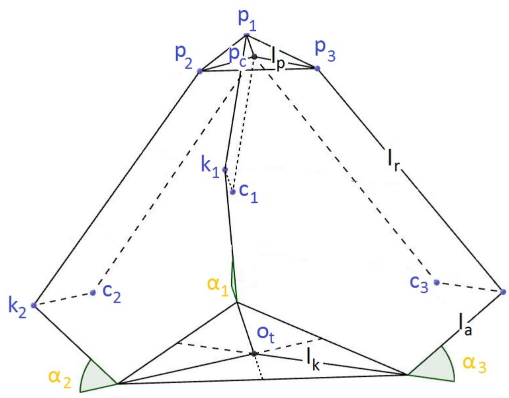


Fig. 1. TAC Geometry.

Table 1. TAC paramters.

| Notation | Unit | Description |
|---|---|---|
| $l_k$ | m | Length from center of the bottom plate to the arm |
| $l_a$ | m | Length of the arm |
| $l_r$ | m | Length of the rod |
| $l_p$ | m | Length from the center of the top plate to the rod |
| $\alpha_i$ | rad | Angle between bottom plate and arm $i$ |
| $u_i$ | rad | input to DC-motors |
| $\mathbf{k}_i$ | - | Knee-point between arm and rod $i$ |
| $\mathbf{c}_i$ | - | Indented Knee-point $i$ |
| $\mathbf{p}_i$ | - | Point connecting rod and top plate |
| $\mathbf{p}_c$ | - | Center-point of the top plate, position of the TCP |
| $i$ | - | Jointed-arm number, $i \in \{1, 2, 3\}$. |

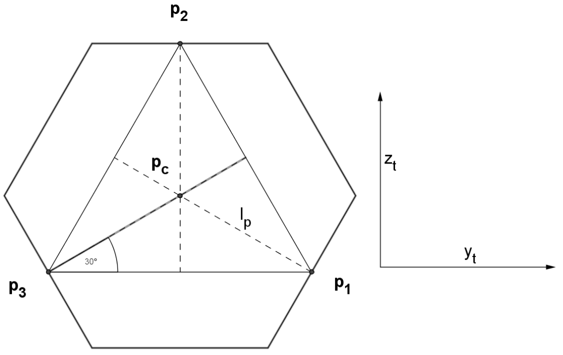


Fig. 2. Colibri top plate seen from above, frame $\{t\}$.

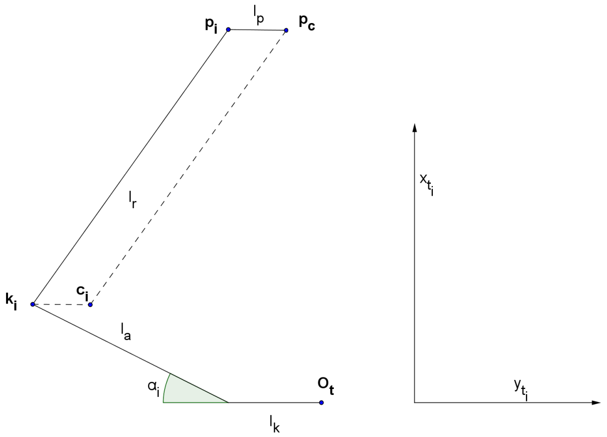


Fig. 3. Colibri arm $i$ seen from one of the sides, frame $\{t_i\}$.

The main frame of orientation, denoted $\{t\}$ with coordinates $(x_t, y_t, z_t)$, is shown in Fig. 2, with the $x$-axis pointing out of the paper plane, $z$-axis up and the $y$-axis to the right. In Fig. 3 the individual frame for any joint is shown, denoted $\{t_i\}$= $(x_{t_i}, y_{t_i}, z_{t_i})$ for $i \in \{1, 2, 3\}$, where the $y$-axis points from the knee into the center. Both {t}- and $\{t_i\}$-frame have the same origin, $\mathbf{o}_t$, thus transforming between between these frames is done by rotations, with the rotation matrices

$$\mathbf{R}^t_{t_1} = \mathbf{R}_{x,\frac{5\pi}{6}}, \quad \mathbf{R}^t_{t_2} = \mathbf{R}_{x,\frac{3\pi}{2}}, \quad \mathbf{R}^t_{t_3} = \mathbf{R}_{x,\frac{\pi}{6}}. \tag{1}$$

The position of the TCP in the different frames is denoted as

$$\mathbf{p}^t_c = [x_c \; y_c \; z_c]^T \text{ and } \mathbf{p}^{t_i}_c = (\mathbf{R}^t_{t_i})^T \mathbf{p}^t_c = [x_{ci} \quad y_{ci} \quad z_{ci}]^T. \tag{2}$$

Each of the three kinematic chains consists of a arm and rod, connected by a knee joint. The position of the knee, $\mathbf{k}_i$, can be derived, when knowing the corresponding angle $\alpha_i$, as

$$\mathbf{k}^t_i = \mathbf{R}^t_{t_i} \left[l_a \cos\alpha_i \;\; -l_k - l_a \sin\alpha_i \;\; 0\right]^T \quad \forall i \in \{1, 2, 3\}, \tag{3}$$

whereas the position of the top plate cannot be found without knowing all three angles. The length of the rod, $l_r$, is constant, a fact that can be exploited to derive the relation between all angles and the TCP. By placing an indented knee-point, $\mathbf{c}_i$, a distance of $l_p$ in the $y_{t_i}$-direction in the $\{t_i\}$-frame yields a point which will be a constant distance of $l_r$ from $\mathbf{p}_c$. $\mathbf{c}_i$ can be described as

$$\mathbf{c}^t_i = \mathbf{R}^t_{t_i} \left[l_a \sin\alpha_i \quad a - l_a \cos\alpha_i \quad 0\right]^T, \quad \forall i \in \{1, 2, 3\}, \tag{4}$$

where $a = l_p - l_k$. The vector $\mathbf{s}^t_i$ is defined as the vector from $\mathbf{c}^t_i$ to $\mathbf{p}^t_c$, i.e.

$$\mathbf{s}^t_i \equiv \mathbf{p}^t_c - \mathbf{c}^t_i \quad \forall i \in \{1, 2, 3\}. \tag{5}$$

$\boldsymbol{p}^t_c$ can be seen as the crossing point of three spheres with radius $l_r$ and center in each indented knee point $\boldsymbol{c}^t_i$, as presented in Fig. 4. With this information, the vector-loop closure equation can be found as

$$\|\mathbf{s}^t_i\|^2_2 = l^2_r \quad \forall i \in \{1, 2, 3\}, \tag{6}$$

where the norm has been squared to avoid the root in the equation. Eq. (6) is a useful tool for describing the system dynamics (Codourey (1988); Williams (2015); Andrioaia et al. (2012)).

## 3. INVERSE POSITION KINEMATICS

The Inverse Position Kinematics (IPK) solution of the system is a way of finding the joint angles, $\boldsymbol{\alpha} = [\alpha_1 \;\; \alpha_2 \;\; \alpha_3]^T$, given the Cartesian coordinates of the TCP, $\mathbf{p}^t_c$ (Williams, 2015). This is done in the $\{t_i\}$-frame, with vector

$$\mathbf{c}^{t_i}_i = [x_i \quad y_i \quad z_i]^T. \tag{7}$$

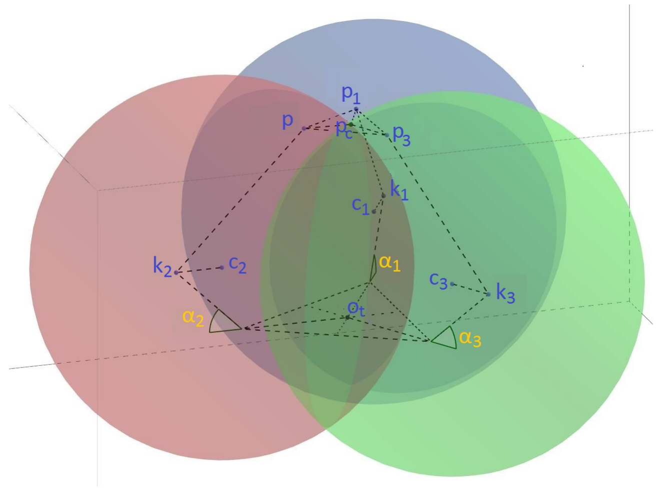


Fig. 4. Tool Center Point $\boldsymbol{p}_c^t$ described by three spheres.

Eq. (6) can be expanded,

$$\|\mathbf{p}_c^{t_i}\|_2^2 + \|\mathbf{c}_i^{t_i}\|_2^2 - l_r^2 - 2(x_{ci}x_i + y_{ci}y_i + z_{ci}z_i) = 0, \tag{8}$$

where

$$\|\mathbf{c}_i^{t_i}\|_2^2 = l_a^2 + a^2 - 2al_a \cos\alpha_i. \tag{9}$$

Eq. (8) is then expanded for all three joints in the individual $\{t_i\}$-frames

$$\begin{aligned}
&\|\mathbf{p}_c^{t_1}\|_2^2 + \|\mathbf{c}_1^{t_1}\|_2^2 - l_r^2 - 2x_{c1}l_a\sin\alpha_1 \\
&\quad + (\sqrt{3}y_{c1} - z_{c1})(a - l_a\cos\alpha_1) = 0, \\
&\|\mathbf{p}_c^{t_2}\|_2^2 + \|\mathbf{c}_2^{t_2}\|_2^2 - l_r^2 - 2x_{c2}l_a\sin\alpha_2 \\
&\quad + 2z_{c2}(a - l_a\cos\alpha_2) = 0, \\
&\|\mathbf{p}_c^{t_3}\|_2^2 + \|\mathbf{c}_3^{t_3}\|_2^2 - l_r^2 - 2x_{c3}l_a\sin\alpha_3 \\
&\quad - (\sqrt{3}y_{c3} + z_{c3})(a - l_a\cos\alpha_3) = 0,
\end{aligned} \tag{10}$$

where the three equations are on the form

$$e_i\cos\alpha_i + f_i\sin\alpha_i + g_i = 0 \quad \forall i \in \{1,2,3\}, \tag{11}$$

with

$$\begin{aligned}
f_1 &= f_2 = f_3 = -2x_{ci}l_a, \\
e_1 &= (-\sqrt{3}y_{c1} + z_{c1} - 2a)l_a, \\
e_2 &= -2(z_{c2} + a)l_a, \\
e_3 &= (\sqrt{3}y_{c3} + z_{c3} + 2a)l_a, \\
g_1 &= \|\mathbf{p}_c^{t_1}\|_2^2 + l_a^2 + a^2 - l_r^2 + (\sqrt{3}y_{c1} - z_{c1})a, \\
g_2 &= \|\mathbf{p}_c^{t_2}\|_2^2 + l_a^2 + a^2 - l_r^2 + 2z_{c2}a, \\
g_3 &= \|\mathbf{p}_c^{t_3}\|_2^2 + l_a^2 + a^2 - l_r^2 - (\sqrt{3}y_{c3} + z_{c3})a.
\end{aligned} \tag{12}$$

Tangent Half-Angle Substitution (Williams, 2015) is a method that can be used to solve (11). By defining a variable $\gamma_i \equiv \tan(\frac{\alpha_i}{2})$, $\cos\alpha_i$ and $\sin\alpha_i$ can be substituted with

$$\cos\alpha_i = \frac{1-\gamma_i^2}{1+\gamma_i^2}, \quad \sin\alpha_i = \frac{2\gamma_i}{1+\gamma_i^2}. \tag{13}$$

Inserting (13) into (11) leads to

$$(g_i - e_i)\gamma_i^2 + (2f_i)\gamma_i + (g_i + e_i) = 0. \tag{14}$$

One can recognize (14) as a second order polynomial which can be solved with the quadratic formula

$$\gamma_{i_{1,2}} = \frac{-f_i \pm \sqrt{e_i^2 + f_i^2 - g_i^2}}{g_i - e_i} \tag{15}$$

This yields two solutions for $\gamma_i$, and two solutions for $\alpha_i$ since $\alpha_i = 2\arctan(\gamma_i)$. One solution has the knees pointing outwards, $\alpha_i < \pi/2$ and the other has the knees pointing inwards, $\alpha_i > \pi/2$. As long as the TCP is inside the workspace, the solutions are well-defined and real. When the solutions are different this will in total yield eight different configurations for the TAC, but the solution chosen should be the one with all knees pointing outward, i.e. satisfying $|\alpha_i| < \frac{\pi}{2}$. The IPK solution can be formed into a function such that

$$\boldsymbol{\alpha} = IPK(\mathbf{p}_c^t). \tag{16}$$

## 4. FORWARD POSITION KINEMATICS

The Forward Position Kinematics (FPK) solution is the inverse of the IPK as it yields $\mathbf{p}_c^t$ given $\boldsymbol{\alpha}$. It can be solved analytically with Gaussian Elimination (Coope, 2000). Eq. (6) can be rewritten as

$$(\boldsymbol{p}_c^t)^T\boldsymbol{p}_c^t - 2(\boldsymbol{p}_c^t)^T\boldsymbol{c}_i^t + (\boldsymbol{c}_i^t)^T\boldsymbol{c}_i^t = l_r^2. \tag{17}$$

One want to remove the quadratic parts of the equation to make it easier to solve. This can be done by introducing the variables

$$r = (\boldsymbol{p}_c^t)^T\boldsymbol{p}_c^t, \quad b_i = (\boldsymbol{c}_i^t)^T\boldsymbol{c}_i^t - l_r^2, \tag{18}$$

which gives

$$(\mathbf{p}_c^t)^T\mathbf{c}_i^t = (r + b_i)/2, \quad \forall i \in \{1,2,3\}. \tag{19}$$

By changing the notation to

$$\mathbf{C} = \begin{bmatrix}\mathbf{c}_1^t & \mathbf{c}_2^t & \mathbf{c}_3^t\end{bmatrix}, \quad \mathbf{1} = [1\ 1\ 1]^T, \quad \mathbf{b} = [b_1\ b_2\ b_3]^T, \tag{20}$$

and introduce the substitution variables

$$\mathbf{u} = \mathbf{C}^{-T}\mathbf{1}, \quad \mathbf{v} = \mathbf{C}^{-T}\mathbf{b}, \tag{21}$$

a solution for $\boldsymbol{p}_c^t$ can be found as

$$\boldsymbol{p}_c^t = (r\mathbf{u} + \mathbf{v})/2. \tag{22}$$

A solution for $\boldsymbol{p}_c^t$ is now presented, but $r$ is still unknown. $r$ can be obtained by inserting (22) into (18)

$$r = (\boldsymbol{p}_c^t)^T\boldsymbol{p}_c^t = \frac{1}{4}(r\mathbf{u} + \mathbf{v})^T(r\mathbf{u} + \mathbf{v}). \tag{23}$$

This can again be rewritten to

$$(\mathbf{u}^T\mathbf{u})r^2 + (2\mathbf{u}^T\mathbf{v} - 4)r + \mathbf{v}^T\mathbf{v} = 0, \tag{24}$$

which can be recognized as a second order polynomial. This can be solved for $r$ as

$$r = \frac{2 - \mathbf{u}^T\mathbf{v} \pm \sqrt{(2 - \mathbf{u}^T\mathbf{v})^2 - (\mathbf{u}^T\mathbf{u})(\mathbf{v}^T\mathbf{v})}}{\mathbf{u}^T\mathbf{u}} \tag{25}$$

If the solution from (25) is inserted in (22) two analytical solution for $\boldsymbol{p}_c^t$ given $\boldsymbol{\alpha}$ appears. To understand why two solutions appears one can look back at Fig. 4 where it can be observed that the three spheres cross each other in two points. One solution is the crossing above the TAC base plate, the other is beneath. The valid solution for the crane head is the one over the TAC base plate i.e satisfying $x_c^t > x_i^t\ \forall i \in \{1,2,3\}$.

## 5. KINEMATIC JACOBIAN

The relationship between the velocity of the TCP, $\dot{\mathbf{p}}_c$, and the joint velocities, $\dot{\boldsymbol{\alpha}}$, can be useful for designing a control system (Codourey (1988); Spong et al. (2005)). This kinematic relationship can be found by differentiating (6) as

$$(\mathbf{s}_i^t)^T\dot{\mathbf{s}}_i^t = 0 \tag{26}$$

where $\dot{\mathbf{s}}_i$ can be found as

$$\dot{\mathbf{s}}_i^t = \dot{\mathbf{p}}_c^t - \mathbf{d}_i^t \dot{\alpha}_i, \tag{27}$$

with

$$\mathbf{d}_i^t = \mathbf{R}_{t_i}^t \left[l_a \cos\alpha_i \;\; l_a \sin\alpha_i \;\; 0\right]^T. \tag{28}$$

Inserting (26) into (27) yields

$$(\mathbf{s}_i^t)^T(\dot{\mathbf{p}}_c^t - \mathbf{d}_i^t \dot{\alpha}_i) = 0. \tag{29}$$

Expanding (29) for $i \in \{1, 2, 3\}$ gives

$$\begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix} \dot{\mathbf{p}}_c - \begin{bmatrix} (\mathbf{s}_1^t)^T\mathbf{d}_1^t & 0 & 0 \\ 0 & (\mathbf{s}_2^t)^T\mathbf{d}_2^t & 0 \\ 0 & 0 & (\mathbf{s}_3^t)^T\mathbf{d}_3^t \end{bmatrix} \dot{\boldsymbol{\alpha}} = 0, \tag{30}$$

which finally becomes

$$\dot{\mathbf{p}}_c^t = \mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha})\dot{\boldsymbol{\alpha}}, \tag{31}$$

where

$$\mathbf{J}(\mathbf{p}_c^t, \boldsymbol{\alpha}) = \begin{bmatrix} (\mathbf{s}_1^t)^T \\ (\mathbf{s}_2^t)^T \\ (\mathbf{s}_3^t)^T \end{bmatrix}^{-1} \begin{bmatrix} (\mathbf{s}_1^t)^T\mathbf{d}_1^t & 0 & 0 \\ 0 & (\mathbf{s}_2^t)^T\mathbf{d}_2^t & 0 \\ 0 & 0 & (\mathbf{s}_3^t)^T\mathbf{d}_3^t \end{bmatrix} \tag{32}$$

is the Jacobian of the system.

## 6. WORKSPACE

The IPK solution described in Section 3 will meet difficulties if the position is outside the reachable area of the TAC, i.e. the workspace. To this end, a check to see if the desired position is inside the workspace is necessary. Since the TAC is to be used for motion compensation, it is imperative that it is reliable. If a desired position is found to be outside of the workspace, additional crane joints (if available) must be utilized, operations must be aborted, or reduced accuracy of the motion compensation must be accepted. A method for finding the workspace can be found with the help of Andrioaia et al. (2012). It is possible to design a TAC based on workspace requirements (Stan et al., 2011) but this will not be treated in this paper.

The first step in finding the workspace is to make a cube containing the limits of the maximum reach of the TCP. The approximated limits stay true to the notion that the top plate can not go through the bottom plate, nor that it can be flipped. Hence, the limits are found to be

$$\begin{aligned} X_{max} &= l_a + l_r \\ X_{min} &= 0 \\ Y_{max} \cong Z_{max} &= l_r + l_p - l_k \\ Y_{min} \cong Z_{min} &= -(l_r + l_p - l_k). \end{aligned} \tag{33}$$

The limits can be combined to form the intervals that contain both feasible and infeasible solutions for $\mathbf{p}_c^t$. These intervals are then split into individual points in three dimensions that can be tested as candidate positions for the TCP. The number of points selected will depend on the spational discretization distance $h$, which acts as the distance between points in one dimensions. This yields the arrays

$$\begin{aligned} \mathbf{x_w} &= [x_{w1}, x_{w2}, ..., x_{wn}] \\ \mathbf{y_w} &= [y_{w1}, y_{w2}, ..., y_{wn}] \\ \mathbf{z_w} &= [z_{w1}, z_{w2}, ..., z_{wn}] \end{aligned} \tag{34}$$

where $x_{w1} = X_{min}$, $x_{wn} = X_{max}$, $y_{w1} = Y_{min}$, $y_{wn} = Y_{max}$, $z_{w1} = Z_{min}$ and $z_{wn} = Z_{max}$ with fixed step size $h$. All the different positions found in these arrays are then checked by testing if the IPK solution is real, and joint angles are inside the design range. The positions that are deemed plausible is kept and serves as the basis for the approximated workspace. The smaller $h$, the higher the resolution of the workspace, with the cost of computational load. The plausible coordinates on the outskirts of the workspace will serve as the edges, and by finding the convex hull of all the allowed coordinates, the limits of the workspace is found. With the limits found, it is possible to check if the desired position is outside the limits while the TAC is running. Appropriate actions for transgressing the limits can be moving the set point back towards the limits (saturation), involving the rest of the crane in the compensation action, or aborting the operation, among other alternatives. This paper will not consider the actions for transgressing the limits further. As it is a cumbersome process to find the limits it should be done as an initializing process, or even done separately and merely entering the limits as parameters to the TAC's controller.

## 7. FULL CRANE KINEMATICS

This section will describe the forward and velocity kinematics of a full crane system, from the Center of Origin (CO) to the TAC. Two more coordinate reference frames are described. The North-East-Down (NED) coordinate system, {n} =$(x_n, y_n, z_n)$ with origin $\mathbf{o}_n$, is defined relative to the Earth's reference ellipsoid, and stays fixed as the ship moves. The body-fixed reference frame ,{b} = $(x_b, y_b, z_b)$ with origin $\mathbf{o}_b$ (CO), is a moving coordinate frame that is fixed to the craft (Fossen, 2011).

Table 2. Notation for marine vessels (SNAME, 1950).

| **DOF** | | **Linear and angular velocities** | **Positions and Euler angles** |
|---|---|---|---|
| 1 | motions x dir (surge) | $u$ | $x$ |
| 2 | motions y dir (sway) | $v$ | $y$ |
| 3 | motions z dir (heave) | $w$ | $z$ |
| 4 | rotation x axis (roll) | $p$ | $\phi$ |
| 5 | rotation y axis (pitch) | $q$ | $\theta$ |
| 6 | rotation z axis (yaw) | $r$ | $\psi$ |

Marine vessels are affected by forces and moments in six Degrees Of Freedom (DOF), which are further described in Table 2. The IMU normally consists of accelerometers and gyroscopes, among other sensors, used to estimate the ship's position and attitude, and the velocities of these, as

$$\begin{aligned} \boldsymbol{\eta} &= [x \;\, y \;\, z \;\, \phi \;\, \theta \;\, \psi]^T, \\ \boldsymbol{\nu} &= [u \;\, v \;\, w \;\, p \;\, q \;\, r]^T. \end{aligned} \tag{35}$$

The IMU measures accelerations, which are integrated to get velocities $\boldsymbol{\nu}$ and position $\boldsymbol{\eta}$. Sensor biases, misalignments and temperature differences will understandably cause drift, especially in the position estimations. To obtain measurements of higher quality the IMU has to be combined with other systems. Combining the IMU with the kinematic equations for the ship results in an inertial navigation system (INS), but the measurements still drifts. By including Global Navigation Satellite System (GNSS) measurements in the computations the INS drift can be removed. Obviously the acceleration measurement quality

Table 3. DH table.

| **Link** | $\boldsymbol{a}_j$ | $\boldsymbol{\alpha}_j$ | $\boldsymbol{d}_j$ | $\boldsymbol{\theta}_j$ |
|---|---|---|---|---|
| 1 | $L_{sx}$ | $\pi$ | $-L_{sz}$ | 0 |
| 2 | 0 | $\frac{\pi}{2}$ | $L_0$ | $\beta_0^*$ |
| 3 | $L_1$ | 0 | 0 | $\beta_1^*$ |
| 4 | $L_2$ | $-\frac{\pi}{2}$ | 0 | $\beta_2^*$ |

depends on the IMU quality while the velocity and position measurement quality depends on the GNSS quality (Fossen, 2011).

The transformation from {b}-frame to {n}-frame can be found as

$$\boldsymbol{T}_b^n = \begin{bmatrix} \boldsymbol{R}_b^n & \boldsymbol{o}_b^n \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{36}$$

where

$$\begin{aligned} \boldsymbol{R}_b^n &= \boldsymbol{R}_{x,\phi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{z,\psi}, \\ \boldsymbol{o}_b^n &= [x \ y \ z]^T. \end{aligned} \tag{37}$$

The forward kinematics of the full crane system can be found by using the DH-convention, with the knowledge of the crane's dimensions and configuration (Spong et al., 2005). The following example, seen in Fig. 5, is considering the transformation from $\boldsymbol{o}_b$ inside the ship, along a three-joint crane system to $\boldsymbol{o}_t$. With the information available in Fig. 5, a DH-table can be made, see Table 3. $\beta_j^*$ are controllable angles. Each row in the DH-table is inserted into (38) to garner the transformation matrix $A_j$ from link $j-1$ to link $j$, and in this example $j = 1, 2, 3, 4$.

$$\begin{aligned} \boldsymbol{A}_j &= \boldsymbol{Rot}_{z,\theta_j}\boldsymbol{Trans}_{z,d_j}\boldsymbol{Trans}_{x,a_j}\boldsymbol{Rot}_{z,\alpha} \\ &= \begin{bmatrix} c_{\theta_j} & -s_{\theta_j}c_{\alpha_j} & s_{\theta_j}s_{\alpha_j} & a_j c_{\theta_j} \\ s_{\theta_j} & c_{\theta_j}c_{\alpha_j} & -c_{\theta_j}s_{\alpha_j} & a_j s_{\theta_j} \\ 0 & s_{\alpha_j} & c_{\alpha_j} & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{38}$$

The notation for $\sin(\cdot)$ and $\cos(\cdot)$ in (38) is $s_x \equiv \sin x$ and $c_x \equiv \cos x$. The transformation matrix between frames can be found as

$$\mathbf{T}_n^0 = \mathbf{A}_0 \cdots \mathbf{A}_n \tag{39}$$

where $n$ is the number of joints. The transformation matrix can be further broken down as

$$\mathbf{T}_n^0 = \begin{bmatrix} \boldsymbol{R}_n^0 & \boldsymbol{o}_n^0 \\ \boldsymbol{0} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_n^0 & \boldsymbol{y}_n^0 & \boldsymbol{z}_n^0 & \boldsymbol{o}_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{40}$$

where $\boldsymbol{R}_n^0$ is a rotation matrix consisting of the three vectors $\boldsymbol{x}_n^0$, $\boldsymbol{y}_n^0$ and $\boldsymbol{z}_n^0$ , and $\boldsymbol{o}_n^0$ is the translation vector.

$$\boldsymbol{T}_t^b = \boldsymbol{T}_4^1 = \boldsymbol{A}_1\boldsymbol{A}_2\boldsymbol{A}_3\boldsymbol{A}_4 = \begin{bmatrix} \boldsymbol{R}_t^b & \boldsymbol{o}_t^b \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{41}$$

The inverse kinematics for the full crane is the method of finding the joint angles, $\boldsymbol{\beta}$, knowing the position of the end-effector, $\boldsymbol{o}_t^b$. By transforming this to the position of the first joint, $\boldsymbol{o}_t^0$ can be found as

$$\boldsymbol{o}_t^0 = \boldsymbol{R}_b^0\boldsymbol{o}_t^b + \boldsymbol{o}_b^0 = \begin{bmatrix} x_b^0 & y_b^0 & z_b^0 \end{bmatrix}. \tag{42}$$

From the first joint and up, the crane can be recognized as the elbow manipulator from (Spong et al., 2005). The angles of this configuration can be found as

$$\begin{aligned} \beta_0 &= \arctan\left(\frac{y_b^0}{x_b^0}\right), \quad \beta_2 = \arctan\left(\frac{\pm\sqrt{1-\delta^2}}{\delta}\right), \\ \beta_1 &= \arctan\left(\frac{z_b^0 - L_0}{\sqrt{(x_b^0)^2 + (y_b^0)^2}}\right) \\ &\quad - \arctan\left(\frac{L_2 \sin\beta_2}{L_1 + L_2\cos\beta_2}\right), \end{aligned} \tag{43}$$

where

$$\delta = \frac{(x_b^0)^2 + (y_b^0)^2 + (z_b^0 - L_0)^2 - L_1^2 - L_2^2}{2L_1L_2}. \tag{44}$$

The two separate solutions for $\beta_2$ give elbow-down or elbow-up configuration of the manipulator, as the positive or negative square root, respectively. For this example elbow-up is desired, as seen in Fig. 5.

The Jacobian matrix, $\mathbf{J}_\beta$, determines the velocity relationship between the end effector, $\boldsymbol{\xi}$, and the joints, $\dot{\boldsymbol{\beta}} = \begin{bmatrix}\dot{\beta}_0 & \dot{\beta}_1 & \dot{\beta}_2\end{bmatrix}^T$. The velocity kinematics can be found as

$$\boldsymbol{\xi}_t^b = \begin{bmatrix} \boldsymbol{v}_t^b \\ \boldsymbol{\omega}_t^b \end{bmatrix} = \mathbf{J}_\beta\dot{\boldsymbol{\beta}} \tag{45}$$

where

$$\mathbf{J}_\beta = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{v_1}...\mathbf{J}_{v_n} \\ \mathbf{J}_{\omega_1}...\mathbf{J}_{\omega_n} \end{bmatrix} \tag{46}$$

in which $\mathbf{J}_v$ is the Linear Velocity Jacobian and $\mathbf{J}_\omega$ is the Angular Velocity Jacobian, where both have a column for every link in Table 3 (Spong et al., 2005). For revolute joints, as in this example, the $i^{th}$ column can be found as

$$\begin{bmatrix} \mathbf{J}_{v_i} \\ \mathbf{J}_{\omega_i} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}), \\ \mathbf{z}_{i-1}. \end{bmatrix}, \tag{47}$$

and since the first link in Table 3 is not movable, this is skipped and the Jacobian matrix can be found as

$$\mathbf{J}_\beta = \begin{bmatrix} \mathbf{z}_1^0 \times (\mathbf{o}_4^0 - \mathbf{o}_1^0) & \mathbf{z}_2^0 \times (\mathbf{o}_4^0 - \mathbf{o}_2^0) & \mathbf{z}_3^0 \times (\mathbf{o}_4^0 - \mathbf{o}_3^0) \\ \mathbf{z}_1^0 & \mathbf{z}_2^0 & \mathbf{z}_3^0 \end{bmatrix}, \tag{48}$$

where all vectors can be found from (39) after calculating the transformation matrices for each link.

## 8. SYSTEM CONTROLLER

The desired position of the TCP, $\boldsymbol{p}_d^n$, can be given by the crane operator, and should be a position that is not affected by the waves. The desired configuration of the TAC should be such that the TCP has the largest freedom of movement in all directions, i.e. in the middle of the workspace, which gives a desired position in the {t}-frame of $\boldsymbol{p}_d^t = [x_d \quad 0 \quad 0]^T$, where $x_d$ can be found by analyzing the workspace area. The crane joints angle can be found by changing $L_2$ in (43) and (44) with $L_{2d} = L_2 + x_d$, as $\boldsymbol{p}_d^t$ only has value along the x-axis.

The position of the TAC changed by the disturbances of the sea, $\boldsymbol{o}_t^n$, can be found as

$$\boldsymbol{o}_t^n = \boldsymbol{R}_b^n\boldsymbol{o}_t^b + \boldsymbol{o}_b^n. \tag{49}$$

To compensate for the vessel's motion, the TCP should be moved equally in the opposite direction of the disturbance, thus the positional reference should be a vector from $\boldsymbol{o}_t^n$ to $\boldsymbol{p}_d^n$ in the {t}-frame, found as

$$\boldsymbol{p}_{ref}^t = \boldsymbol{R}_n^t\boldsymbol{p}_{ref}^n, \tag{50}$$

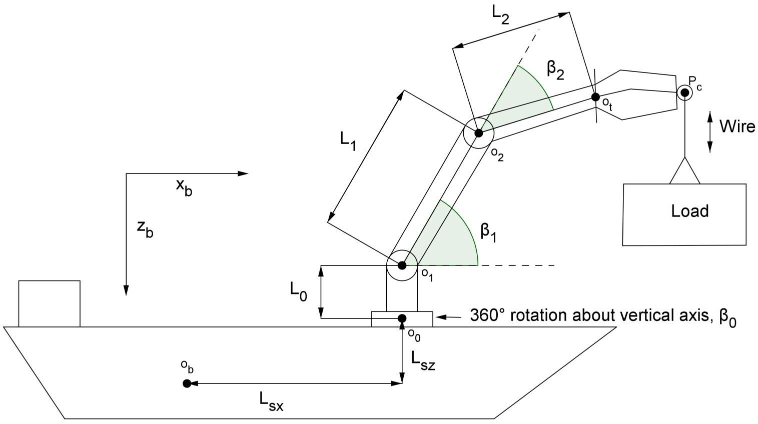


Fig. 5. Ship with full crane system: An elbow manipulator (Spong et al., 2005) and the TAC.

where $\boldsymbol{R}^t_n = (\boldsymbol{R}^b_t)^T(\boldsymbol{R}^n_b)^T$ and $\boldsymbol{p}^n_{ref} = \boldsymbol{p}^n_d - \boldsymbol{o}^n_t$. After checking if $\boldsymbol{p}^t_{ref}$ is inside the workspace, the reference angles, $\boldsymbol{\alpha}_{ref}$, can be found with the use of the IPK function found in Section 3, such that

$$\boldsymbol{\alpha}_{ref} = IPK(\boldsymbol{p}^t_{ref}). \tag{51}$$

The velocity reference can be found by differentiating (50)

$$\begin{aligned}\mathbf{v}^t_{ref} &= \dot{\mathbf{p}}^t_{ref} \\ &= \dot{\mathbf{R}}^t_n \mathbf{p}^n_{ref} + \mathbf{R}^t_n \dot{\mathbf{p}}^n_{ref},\end{aligned} \tag{52}$$

where

$$\begin{aligned}\dot{\mathbf{R}}^t_n &= (\dot{\boldsymbol{R}^b_t})^T(\boldsymbol{R}^n_b)^T + (\boldsymbol{R}^b_t)^T(\dot{\boldsymbol{R}^n_b})^T \\ &= -(\mathbf{S}(\boldsymbol{\omega}^b_t)(\mathbf{R}^b_t)^T + (\mathbf{R}^b_t)^T\mathbf{S}(\boldsymbol{\omega}^n_b))(\mathbf{R}^n_b)^T\end{aligned} \tag{53}$$

and

$$\begin{aligned}\dot{\mathbf{p}}^n_{ref} &= \dot{\mathbf{R}}^b_t \boldsymbol{p}^t_d + \dot{\mathbf{o}}^b_t - \dot{\mathbf{R}}^n_b \mathbf{o}^b_t - \mathbf{R}^n_b \dot{\mathbf{o}}^b_t - \dot{\mathbf{o}}^n_b \\ &= \mathbf{S}(\boldsymbol{\omega}^b_t)\mathbf{R}^b_t \boldsymbol{p}^t_d - \mathbf{S}(\boldsymbol{\omega}^n_b)\mathbf{R}^n_b \mathbf{o}^b_t + (\mathbf{I} - \mathbf{R}^n_b)\mathbf{v}^b_t - \mathbf{v}^n_b\end{aligned} \tag{54}$$

with

$$\begin{aligned}\mathbf{v}^n_b &= [u \ v \ w]^T , \\ \boldsymbol{\omega}^n_b &= [p \ q \ r]^T .\end{aligned} \tag{55}$$

$\boldsymbol{\omega}^b_t$ and $\mathbf{v}^b_t$ are found from (45). The TCP positional velocity is translated to the joints angular velocity by use of the Jacobian matrix from Section 5,

$$\dot{\boldsymbol{\alpha}}_{ref} = \mathbf{J}^{-1}(\mathbf{p}^t_{ref}, \boldsymbol{\alpha}_{ref})\mathbf{v}^t_{ref}. \tag{56}$$

With error dynamics as $\tilde{\boldsymbol{\alpha}} \equiv \boldsymbol{\alpha}_{ref} - \boldsymbol{\alpha}$ and $\dot{\tilde{\boldsymbol{\alpha}}} \equiv \dot{\boldsymbol{\alpha}}_{ref} - \dot{\boldsymbol{\alpha}}$, the input to the DC-motors, $\mathbf{u}_\alpha = [u_1 \ u_2 \ u_3]$, can be chosen with the use of a PID controller,

$$\mathbf{u}_\alpha = \mathbf{K}_p\tilde{\boldsymbol{\alpha}} + \mathbf{K}_i \int \tilde{\boldsymbol{\alpha}}(t)dt + \mathbf{K}_d\dot{\tilde{\boldsymbol{\alpha}}}, \tag{57}$$

where the controller gains are chosen from regulator tuning.

## 9. EXPERIMENTAL SETUP

The experimental setup consists of the controller described in Section 8, a TAC simulator and a simulated supply vessel with an IMU. An illustration of this is provided in Figure 6.

In the simulator the DC motors used to manipulate $\boldsymbol{\alpha}$ are modeled as the first order system

$$\frac{\boldsymbol{\alpha}}{\mathbf{u}_\alpha}(s) = \frac{K_{DC}}{T_{DC}s + 1}, \tag{58}$$

where values for the motor constants $K_{DC}$ and $T_{DC}$ can be found in Table 4. Inverse Laplace transformation and inserting this equation into (31) yields

$$\dot{\mathbf{p}}^t_c = \mathbf{J}(\mathbf{p}^t_c, \boldsymbol{\alpha})(-\frac{1}{T_{DC}}\boldsymbol{\alpha} + \frac{K_{DC}}{T_{DC}}\mathbf{u}_\alpha), \tag{59}$$

which represents the TAC system dynamics. When designing the ship and simulating the IMU data used in this simulator, the MSS toolbox (Fossen, 2011) proved very useful. A supply ship model that takes a controllable wave spectrum as input, and outputs $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ was created from the MSS toolbox. In this experiment measurement noise was neglected as it is not the focus of this paper. Essential parameter values used in the experimental setup can be found in Table 4. The simulator also includes the motion of the full crane, where a path could be generated for the TCP in a pattern such as loading or offloading, and then finding the joint angles through inverse kinematics.

Table 4. Parameter values.

| **Parameter** | **Value** | Unit |
|---|---|---|
| $l_k$ | 2.00 | m |
| $l_a$ | 3.25 | m |
| $l_r$ | 7.00 | m |
| $l_p$ | 1.25 | m |
| $\alpha_{min}$ | -40 | deg |
| $\alpha_{max}$ | 70 | deg |
| $T_{DC}$ | 0.2 | s |
| $K_{DC}$ | 1 | - |
| $K_p$ | 10 | - |
| $K_i$ | 60 | - |
| $K_d$ | 1 | - |
| $L_{sx}$ | -20.70 | m |
| $L_{sz}$ | 3.00 | m |
| $L_0$ | 3.00 | m |
| $L_1$ | 19.20 | m |
| $L_2$ | 9.60 | m |

## 10. SIMULATION RESULTS

The conducted experiment used compensation action with only the TAC, with positive results for significant wave height, $H_s$, up to and including 3.0 meters. This puts the sea state to 5, meaning rough sea (Fossen, 2011). As the focus of this paper is the TAC, the crane is in this simulation placed in a stationary configuration, with

$$\beta_0 = -\frac{\pi}{2}, \quad \beta_1 == \frac{\pi}{4}, \quad \text{and } \beta_2 = -\frac{\pi}{4}, \tag{60}$$

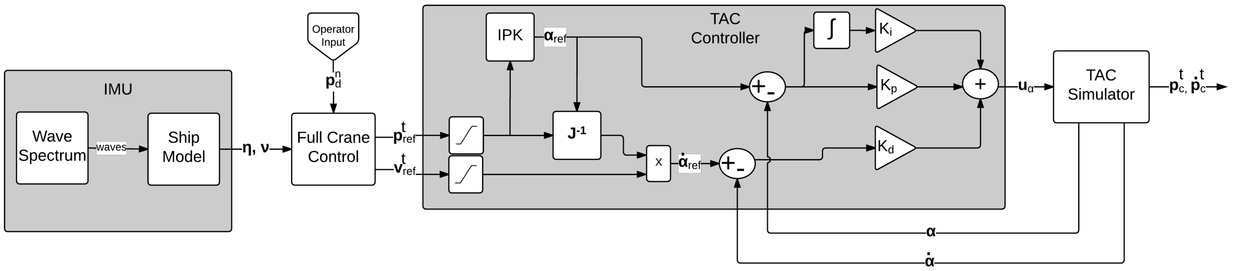


Fig. 6. Block diagram representation of the experimental setup.

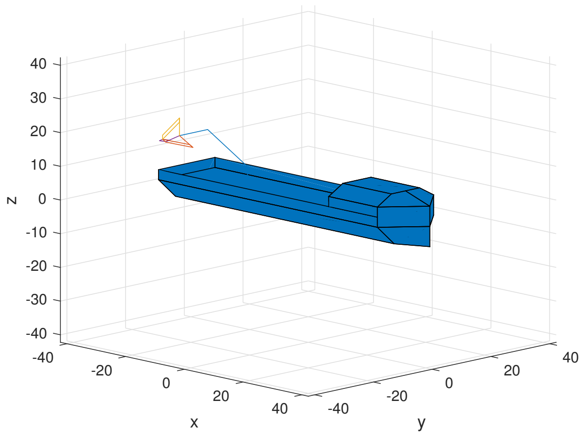


Fig. 7. 3D rendering ship, crane and TAC.

such that the crane points to the starboard side of the ship, with the TAC in parallel with the ship deck, as seen in Fig. 7. The figure also demonstrates the dimensions of the system.

In Fig. 8 the 6 DOF's simulated by the ship simulator can be seen. Fig. 9 shows the how the control loop works with $\mathbf{p}_c^t$ and $\mathbf{p}_{ref}^t$, whereas Fig. 10 shows this in {n}-frame, and compares $\mathbf{p}_c^n$ to the uncompensated TCP position $\mathbf{p}_s^n$. Fig. 11 shows the position error of the compensation, $e_c$, compared to the error of the uncompensated case, $e_s$. Fig. 12 shows $\dot{\mathbf{p}}_c^t$ during the testing, and Fig. 13 shows the inputs $\boldsymbol{u}_\alpha$ along with the corresponding $\boldsymbol{\alpha}$. The controller has been tuned such that the position follows the reference closely, as can be seen in Fig. 9.

## 11. DISCUSSION AND CONCLUSION

The goal of this paper is to show that the TAC can be used for motion compensation, and to derive the equations and algorithms needed. Because of the TAC's highly nonlinear nature, this is not a straightforward task, and the solution requires several steps. Controllers for the compensation using merely the TAC has been developed, and shown to be successful for rough sea. With $H_s$ higher than 3.0 meters the workspace limits were reached, which reduces the compensation's efficiency. The TAC provides fast and accurate motion compensation inside its workspace limits. It is reasonable to expect good motion compensation in higher sea if the rest of the crane is also included in the control loop.

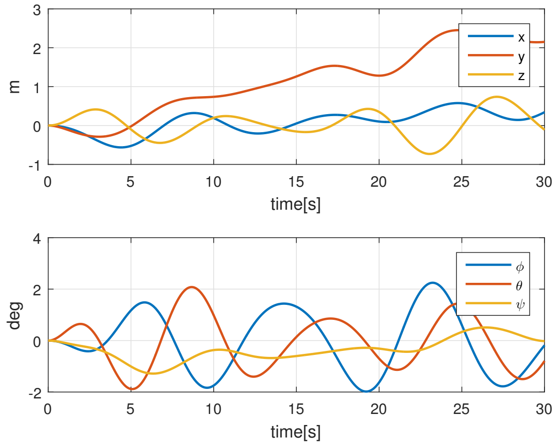


Fig. 8. Simulated vessel position and orientation $\boldsymbol{\eta}$ in {n}-frame.

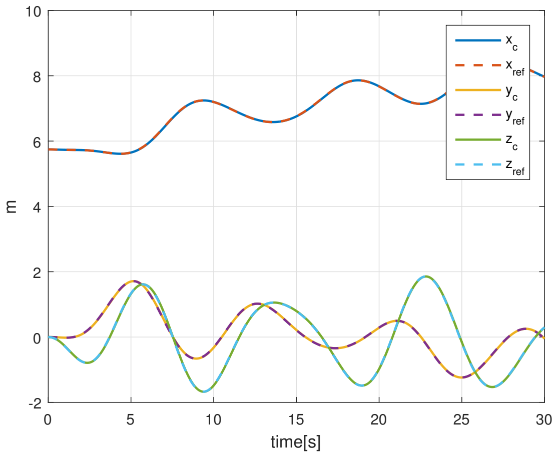


Fig. 9. Simulated TCP position $\boldsymbol{p}_c^t$.

The maximum sea state, where the crane can have full mobility, should be determined if it is to be the only compensating force on the cargo. Since the crane's job ultimately is to get its cargo safe and sound to its destination, and not necessarily to keep the TCP still, the workspace requirements might differ and the model must be upgraded to include the suspended load.

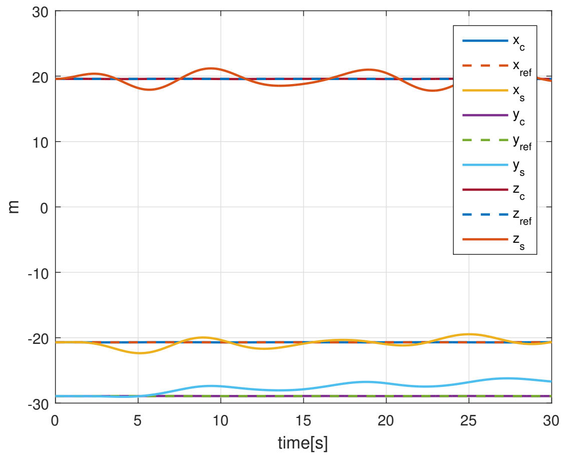


Fig. 10. Simulated TCP position $\boldsymbol{p}_c^n$.

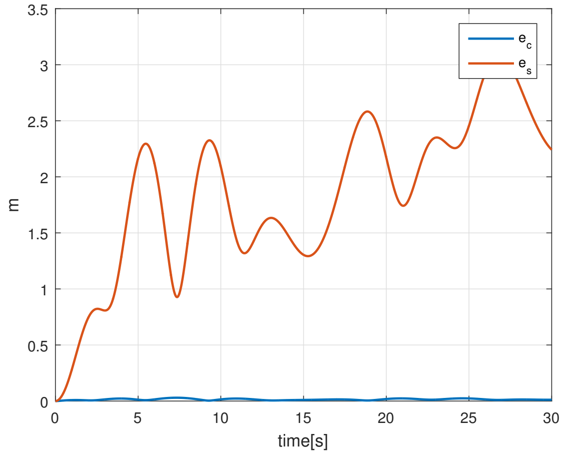


Fig. 11. Error compensated versus uncompensated.

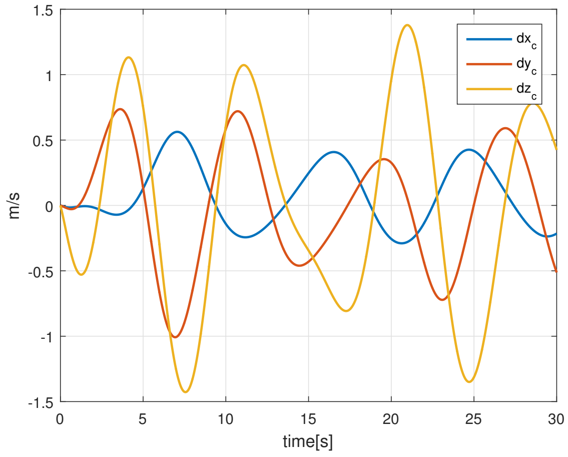


Fig. 12. Simulated TCP velocity $\dot{\boldsymbol{p}}_c^t$.

REFERENCES

Andrioaia, D., Pascu, M., Mihaila, L., and Obrea, C.F. (2012). Determining the workspace in case of the robots with parallel structure delta 3dof. In *Annals & Proceedings of DAAAM International 2012*, volume 23. DAAAM International.

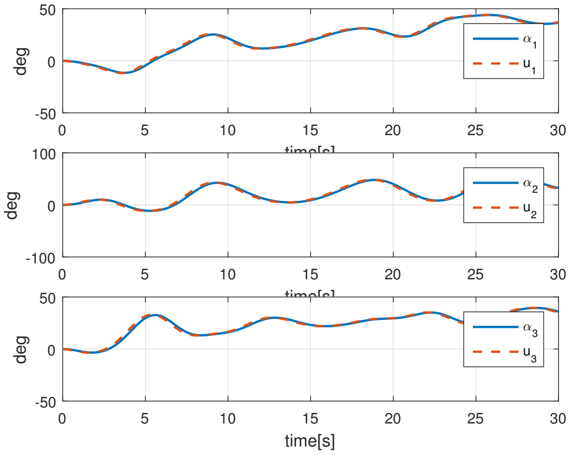


Fig. 13. Simulated angles $\boldsymbol{\alpha}$ and input $\boldsymbol{u}_\alpha$.

Clavel, R. (1988). Delta, a fast robot with parallel geometry. In *1988 18th International Symposium on Industrial Robots (ISIR) in Lausanne, Switzerland*, 91–100. Berlin: Springer-Verlag.

Codourey, A. (1988). Dynamic modeling of parallel robots for computed-torque control implementation. *The International Journal of Robotics Research*, 17, 1325–1336.

Coope, I.D. (2000). Reliable computation of the points of intersection of $n$ spheres in $\mathbb{R}^n$. *ANZIAM Journal*, 461–477.

Fang, Y., Wang, P., Sun, N., and Zhang, Y. (2014). Dynamics analysis and nonlinear control of an offshore boom crane. *IEEE Transactions on Industrial Electronics*, 61(1), 414–427.

Fossen, T.I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley.

Johansen, T.A., Fossen, T.I., Sagatun, S.I., and Nielsen, F.G. (2003). Wave synchronizing crane control during water entry in offshore moonpool operations - experimental results. *IEEE Journal of Oceanic Engineering*, 28(4), 720–728.

Kchler, S., Mahl, T., Neupert, J., Schneider, K., and Sawodny, O. (2011). Active control for an offshore crane using prediction of the vessel's motion. *IEEE/ASME Transactions on Mechatronics*, 16(2), 297–309.

Laribi, M.A., Romdhane, L., and Zeghloul, S. (2008). Advanced Synthesis of the DELTA Parallel Robot for a Specified Workspace. *Parallel Manipulators Towards New Applications*, 207–210.

Messineo, S. and Serrano, A. (2009). Offshore crane control based on adaptive external models. *Automatica*, 45(11), 2546 – 2556.

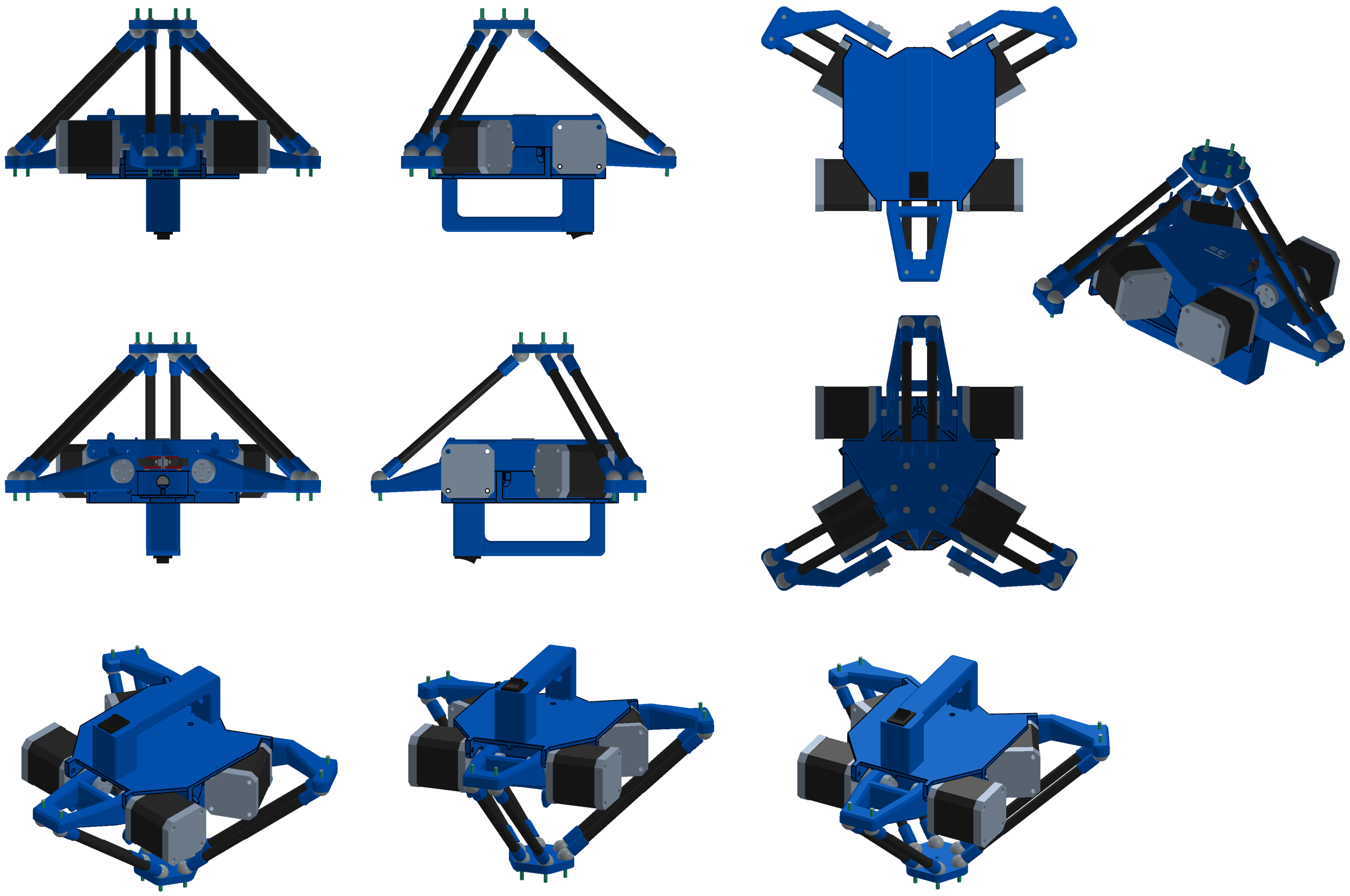
SNAME (1950). The society of naval architects and marine engineers. nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin No. 15.*

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*. John Wiley & Sons.

Stan, S.D., Manic, M., Szep, C., and Balan, R. (2011). Performance analysis of 3 dof delta parallel robot. In *2011 4th International Conference on Human System Interactions (HSI)*, 215–220. IEEE.

Williams, R.L. (2015). The delta parallel robot: Kinematics solutions. *Mechanical Engineering, Ohio University.*

# Appendix B
# Prototype Solidworks Sketch



SOLIDWORKS Student Edition.
For Academic Use Only.

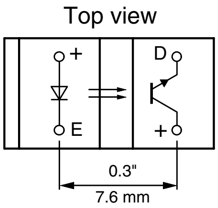
| | | | | | | |
|---|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED:<br>DIMENSIONS ARE IN MILLIMETERS<br>SURFACE FINISH:<br>TOLERANCES:<br>LINEAR:<br>ANGULAR: | FINISH: | | | DEBURR AND<br>BREAK SHARP<br>EDGES | DO NOT SCALE DRAWING | REVISION |
| | NAME | SIGNATURE | DATE | | TITLE: | |
| DRAWN | | | | | | |
| CHK'D | | | | | | |
| APPV'D | | | | | | |
| MFG | | | | | | |
| Q.A | | | | MATERIAL: | DWG NO.<br>Colibri scale model | A0 |
| | | | | WEIGHT: | SCALE:1:1 | SHEET 1 OF 1 |

Appendix C

# Prototype Component Sheets

# Transmissive Optical Sensor with Phototransistor Output

19180_4



19180_3

## DESCRIPTION

The TCST2103, TCST2202, and TCST2300 are transmissive sensors that include an infrared emitter and phototransistor, located face-to-face on the optical axes in a leaded package which blocks visible light. These part numbers include options for aperture width.

## FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 24.5 x 6.3 x 10.8
- Gap (in mm): 3.1
- Typical output current under test: $I_C$ = 4 mA (TCST2103)
- Typical output current under test: $I_C$ = 2 mA (TCST2202)
- Typical output current under test: $I_C$ = 0.5 mA (TCST2300)
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Pb-free

e4
RoHS
COMPLIANT

## APPLICATIONS

- Optical switch
- Photo interrupter
- Counter
- Encoder

## PRODUCT SUMMARY

| PART NUMBER | GAP WIDTH (mm) | APERTURE WIDTH (mm) | TYPICAL OUTPUT CURRENT UNDER TEST [1] (mA) | DAYLIGHT BLOCKING FILTER INTEGRATED |
|---|---|---|---|---|
| TCST2103 | 3.1 | 1 | 4 | Yes |
| TCST2202 | 3.1 | 0.5 | 2 | Yes |
| TCST2300 | 3.1 | 0.25 | 0.5 | Yes |

**Note**

[1] Conditions like in table basic characteristics/coupler

## ORDERING INFORMATION

| ORDERING CODE | PACKAGING | VOLUME [1] | REMARKS |
|---|---|---|---|
| TCST2103 | Tube | MOQ: 1020 pcs, 85 pcs/tube | With mounting flange |
| TCST2202 | Tube | MOQ: 1020 pcs, 85 pcs/tube | With mounting flange |
| TCST2300 | Tube | MOQ: 1020 pcs, 85 pcs/tube | With mounting flange |

**Note**

[1] MOQ: minimum order quantity

## ABSOLUTE MAXIMUM RATINGS [1]

| PARAMETER | TEST CONDITION | SYMBOL | VALUE | UNIT |
|---|---|---|---|---|
| **COUPLER** | | | | |
| Total power dissipation | $T_{amb} \leq 25$ °C | $P_{tot}$ | 250 | mW |
| Ambient temperature range | | $T_{amb}$ | - 55 to + 85 | °C |
| Storage temperature range | | $T_{stg}$ | - 55 to + 100 | °C |
| Soldering temperature | Distance to package: 2 mm; t ≤ 5 s | $T_{sd}$ | 260 | °C |

| ABSOLUTE MAXIMUM RATINGS [1] | | | | |
|---|---|---|---|---|
| PARAMETER | TEST CONDITION | SYMBOL | VALUE | UNIT |
| INPUT (EMITTER) | | | | |
| Reverse voltage | | $V_R$ | 6 | V |
| Forward current | | $I_F$ | 60 | mA |
| Forward surge current | $t_p \leq 10$ µs | $I_{FSM}$ | 3 | A |
| Power dissipation | $T_{amb} \leq 25$ °C | $P_V$ | 100 | mW |
| Junction temperature | | $T_j$ | 100 | °C |
| OUTPUT (DETECTOR) | | | | |
| Collector emitter voltage | | $V_{CEO}$ | 70 | V |
| Emitter collector voltage | | $V_{ECO}$ | 7 | V |
| Collector peak current | $t_p/T = 0.5$, $t_p \leq 10$ ms | $I_{CM}$ | 200 | mA |
| Power dissipation | $T_{amb} \leq 25$ °C | $P_V$ | 150 | mW |
| Junction temperature | | $T_j$ | 100 | °C |

**Note**

[1] $T_{amb} = 25$ °C, unless otherwise specified

## ABSOLUTE MAXIMUM RATINGS

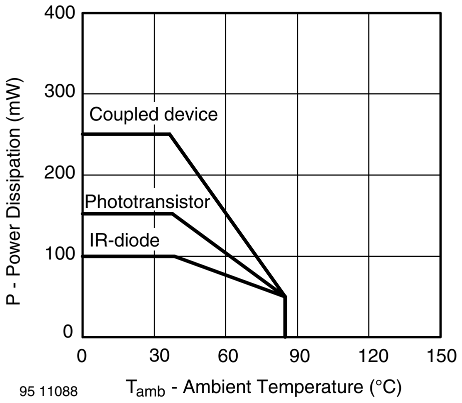


Fig. 1 - Power Dissipation Limit vs. Ambient Temperature

| BASIC CHARACTERISTICS [1] | | | | | | | |
|---|---|---|---|---|---|---|---|
| PARAMETER | TEST CONDITION | PART | SYMBOL | MIN. | TYP. | MAX. | UNIT |
| COUPLER | | | | | | | |
| Current transfer ratio | $V_{CE} = 5$ V, $I_F = 20$ mA | TCST2103 | CTR | 10 | 20 | | % |
| | | TCST2202 | CTR | 5 | 10 | | % |
| | | TCST2300 | CTR | 1.25 | 2.5 | | % |
| Collector current | $V_{CE} = 5$ V, $I_F = 20$ mA | TCST2103 | $I_C$ | 2 | 4 | | mA |
| | | TCST2202 | $I_C$ | 1 | 2 | | mA |
| | | TCST2300 | $I_C$ | 0.25 | 0.5 | | mA |
| Collector emitter saturation voltage | $I_F = 20$ mA, $I_C = 1$ mA | TCST2103 | $V_{CEsat}$ | | | 0.4 | V |
| | $I_F = 20$ mA, $I_C = 0.5$ mA | TCST2202 | $V_{CEsat}$ | | | 0.4 | V |
| | $I_F = 20$ mA, $I_C = 0.1$ mA | TCST2300 | $V_{CEsat}$ | | | 0.4 | V |
| Resolution, path of the shutter crossing the radiant sensitive zone | $I_{Crel} = 10$ % to 90 % | TCST2103 | s | | 0.6 | | mm |
| | | TCST2202 | s | | 0.4 | | mm |
| | | TCST2300 | s | | 0.2 | | mm |

| BASIC CHARACTERISTICS [1] | | | | | | | |
|---|---|---|---|---|---|---|---|
| PARAMETER | TEST CONDITION | PART | SYMBOL | MIN. | TYP. | MAX. | UNIT |
| INPUT (EMITTER) | | | | | | | |
| Forward voltage | $I_F$ = 60 mA | | $V_F$ | | 1.25 | 1.6 | V |
| Junction capacitance | $V_R$ = 0 V, f = 1 MHz | | $C_j$ | | 50 | | pF |
| OUTPUT (DETECTOR) | | | | | | | |
| Collector emitter voltage | $I_C$ = 1 mA | | $V_{CEO}$ | 70 | | | V |
| Emitter collector voltage | $I_E$ = 10 µA | | $V_{ECO}$ | 7 | | | V |
| Collector dark current | $V_{CE}$ = 25 V, $I_F$ = 0 A, E = 0 lx | | $I_{CEO}$ | | | 100 | nA |
| SWITCHING CHARACTERISTICS | | | | | | | |
| Turn-on time | $I_C$ = 2 mA, $V_S$ = 5 V,<br>$R_L$ = 100 Ω (see figure 2) | | $t_{on}$ | | 10 | | µs |
| Turn-off time | $I_C$ = 2 mA, $V_S$ = 5 V,<br>$R_L$ = 100 Ω (see figure 2) | | $t_{off}$ | | 8 | | µs |

**Note**

[1] $T_{amb}$ = 25 °C, unless otherwise specified

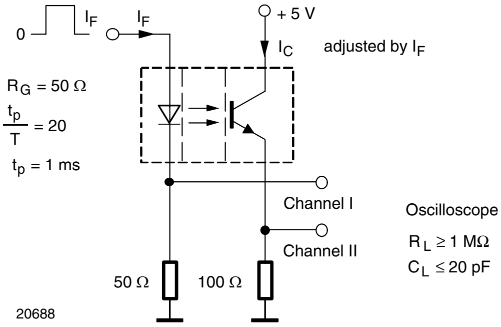


Fig. 2 - Test Circuit for $t_{on}$ and $t_{off}$

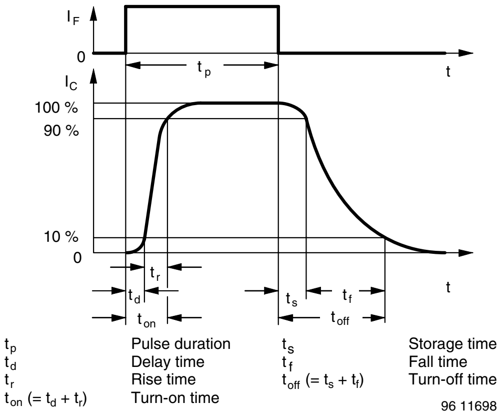


Fig. 3 - Switching Times

## BASIC CHARACTERISTICS

$T_{amb}$ = 25 °C, unless otherwise specified

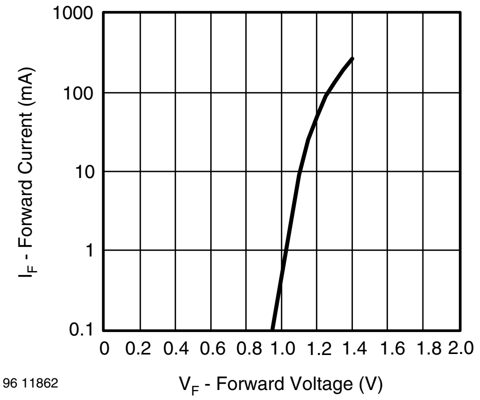


Fig. 4 - Forward Current vs. Forward Voltage

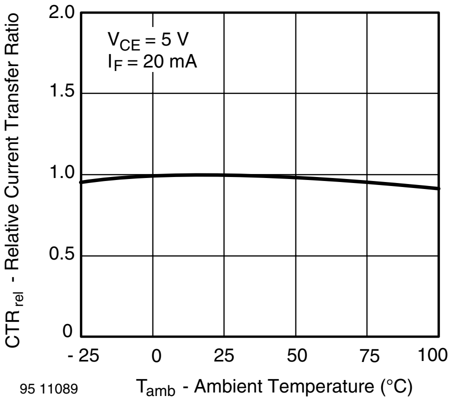


Fig. 5 - Relative Current Transfer Ratio vs. Ambient Temperature

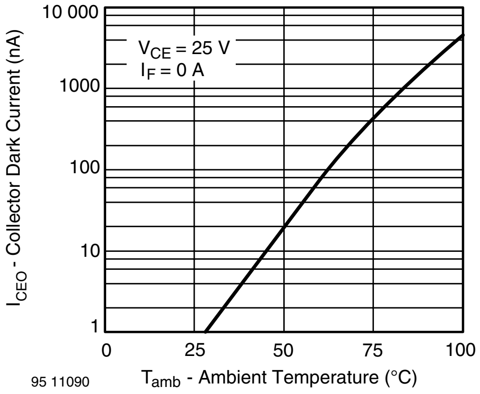


Fig. 6 - Collector Dark Current vs. Ambient Temperature

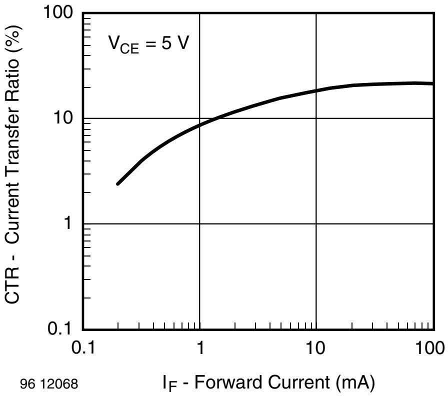


Fig. 9 - Current Transfer Ratio vs. Forward Current

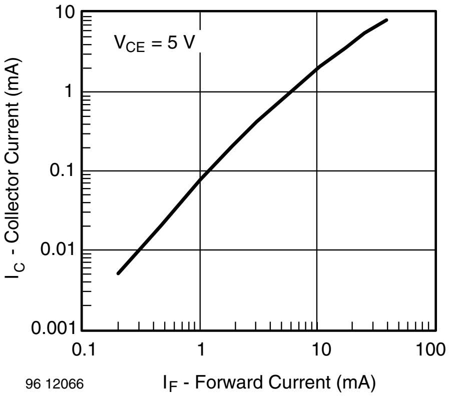


Fig. 7 - Collector Current vs. Forward Current

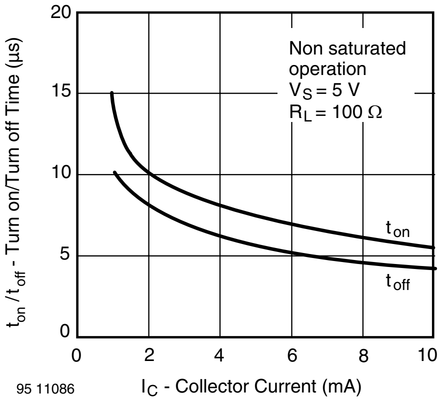


Fig. 10 - Turn-off/Turn-on Time vs. Collector Current

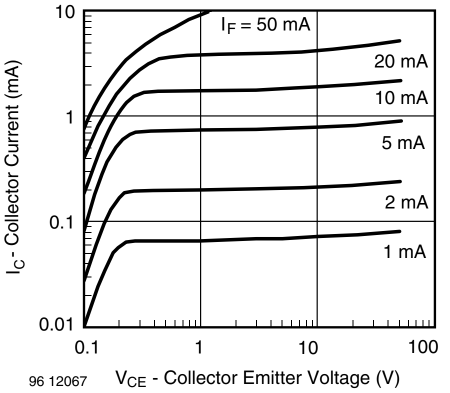


Fig. 8 - Collector Current vs. Collector Emitter Voltage

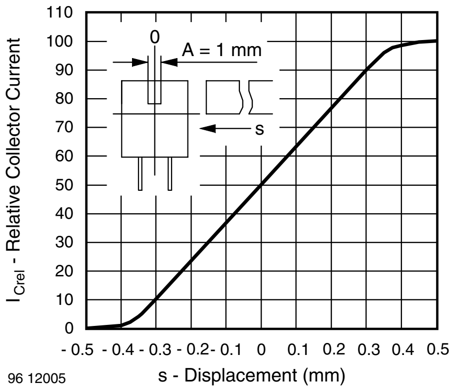
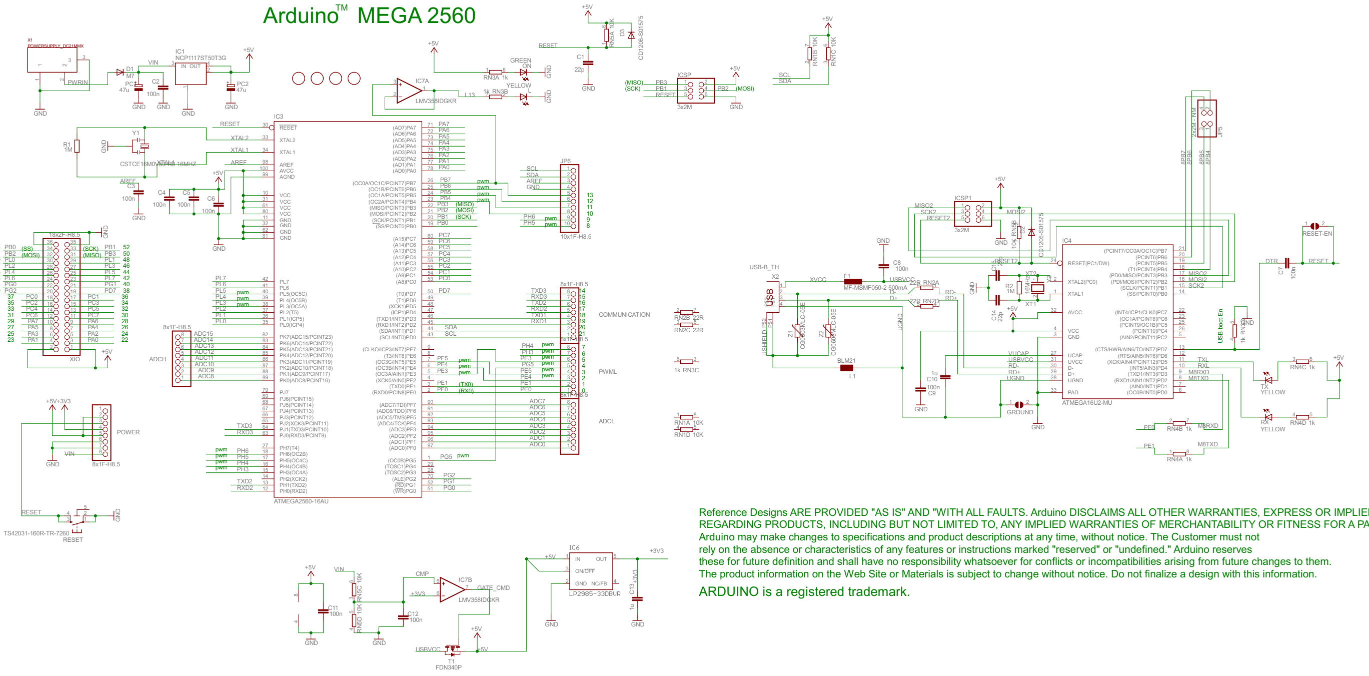


Fig. 11 - Relative Collector Current vs. Displacement

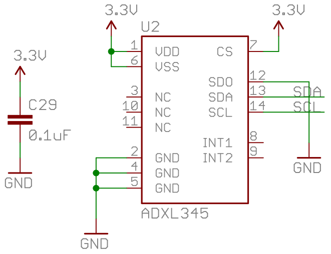
# Arduino™ MEGA 2560



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

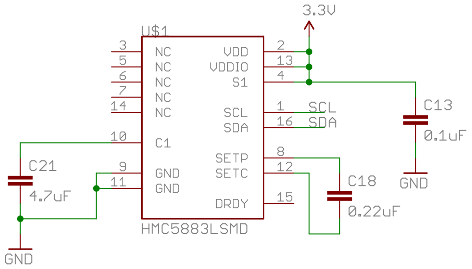
ARDUINO is a registered trademark.

## ACCEL



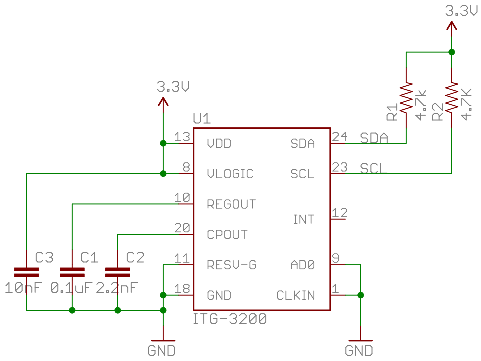
0xA6 for write, 0xA7 for read

## MAG



Released under the Creative Commons Attribution Share-Alike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0
Design by:

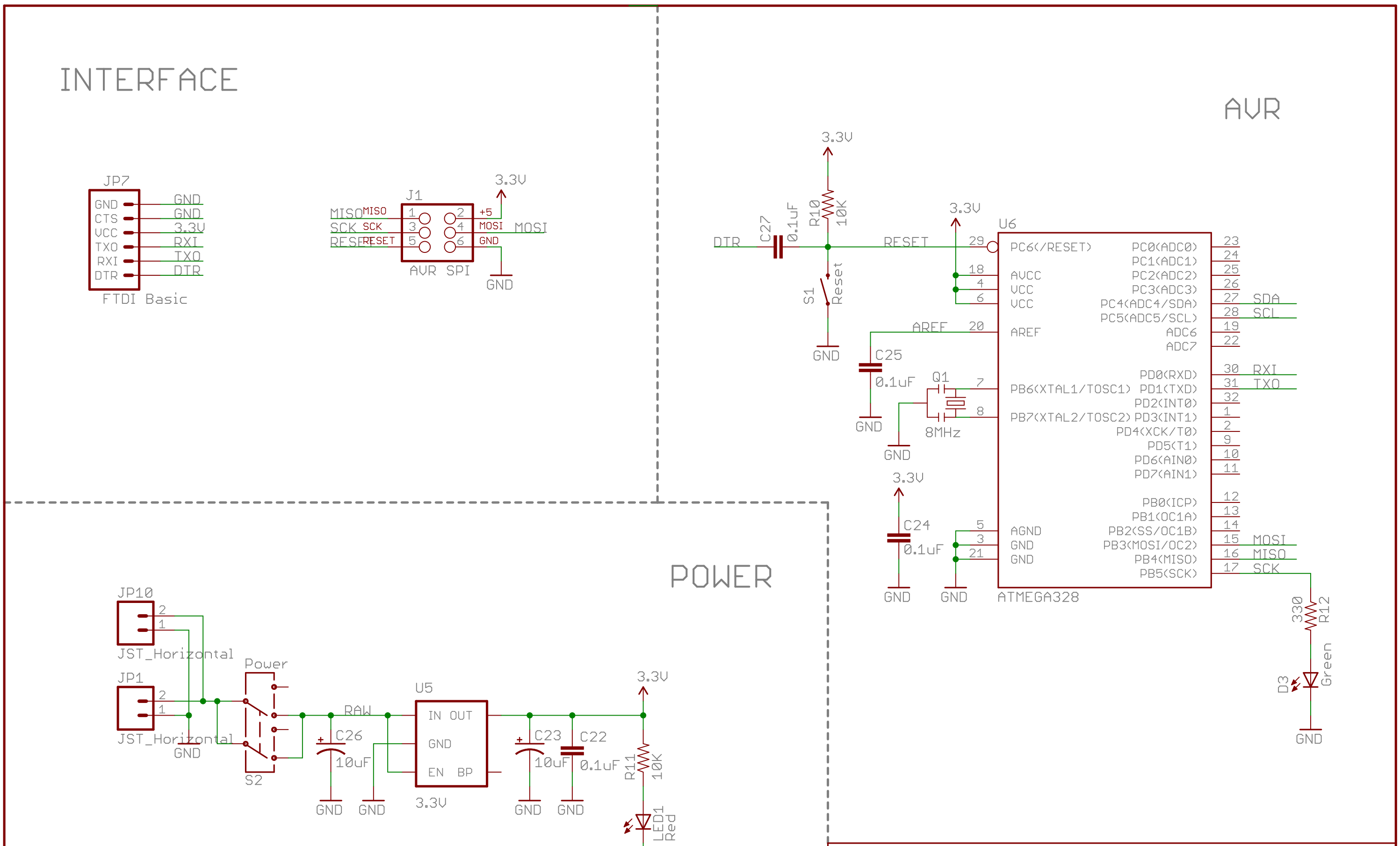
## GYRO



0xD0 for write, 0xD1 for read

open hardware

| | |
|---|---|
| TITLE: 9DOF-Razor-v22 | SFE |
| Document Number: | REV: |
| Date: 6/1/2011 4:05:35 PM | Sheet: 1/2 |



Released under the Creative Commons Attribution Share-Alike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0
Design by:

TITLE: 9DOF-Razor-v22

| Document Number: | REV: |
|---|---|
| Date: 6/1/2011 4:05:35 PM | Sheet: 2/2 |

# RAMPS 1.4

http://reprap.org/wiki/ramps

This is open hardware: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This hardware design is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program.  If not, see <http://www.gnu.org/licenses/>.

**!!! Reversing input power, and/or inserting stepper drivers incorrectly will destroy electronics and cause fire hazard!!!**

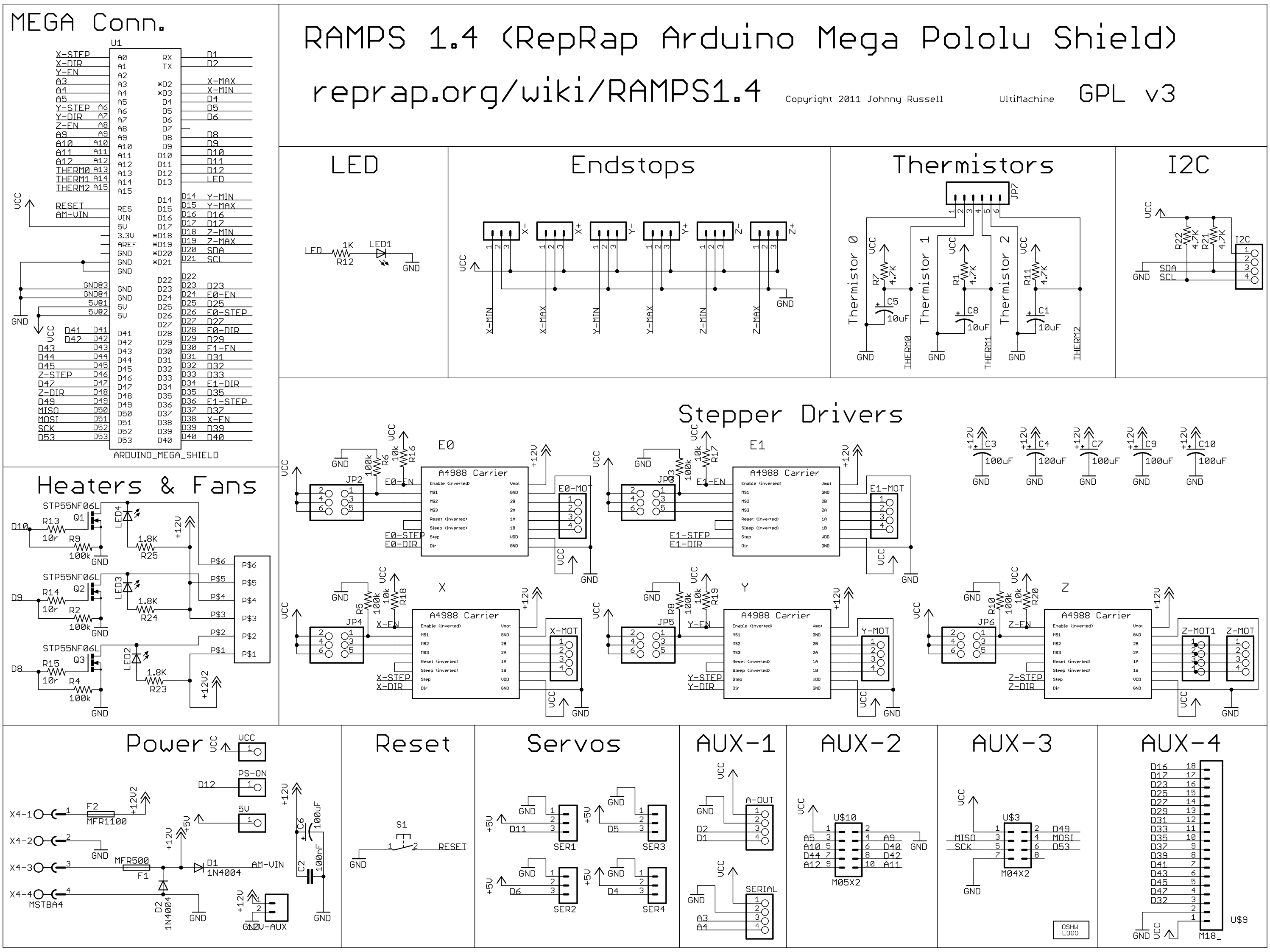
**Test all electronics thoroughly before placing into service.**

**Do not leave power supplied to electronics unattended, or run machines unattended due to the risk of fire and malfunction.**

**This is NOT a toy and it contains small and sharp parts. Children can choke or suffocate by swallowing small objects. Keep all parts away from children and never leave printer/parts unattended.**

**Bill of Materials**

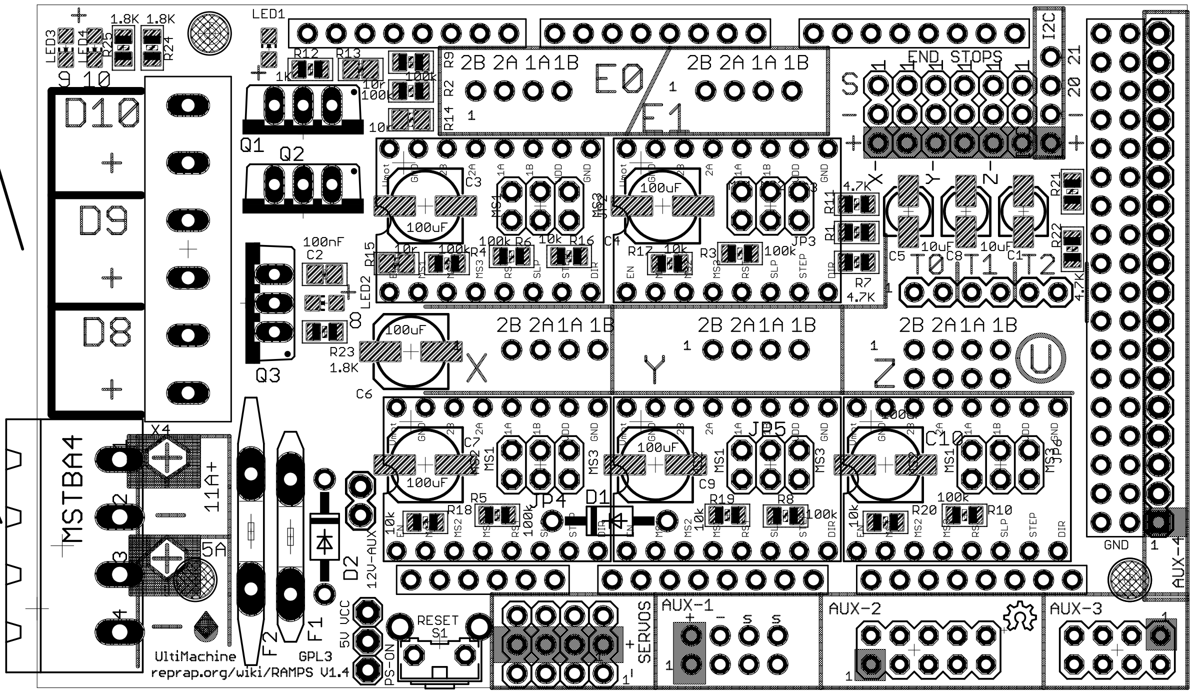
| Qty | Part Name | Value | Package |
|---|---|---|---|
| 3 | C1,C5,C8 | 10uF | 153CLV-0405 |
| 1 | C2 | 100nF | C0805 |
| 6 | C3,C4,C6,C7,C9,C10 | 100uF | 153CLV-0605 |
| 2 | D1,D2 | 1N4004 | DO41-10 |
| 1 | F1 | MFR500 | MFR500 |
| 1 | F2 | MFR1100 | MFR1100 |
| 1 | LED1 | green | CHIP-LED0805 |
| 3 | LED2,LED3,LED4 | red | CHIP-LED0805 |
| 3 | Q1,Q2,Q3 | STP55NF06L | TO220BV |
| 5 | R1,R7,R11,R21,R22 | 4.7K (1%) | R0805 |
| 8 | R2,R3,R4,R5,R6,R8,R9,R | 100k | R0805 |
| 1 | R12 | 1K | R0805 |
| 3 | R23,R24,R26 | 1.8K | R0806 |
| 3 | R13,R14,R15 | 10r | R0805 |
| 5 | R16,R17,R18,R19,R20 | 10k | R0805 |
| 1 | S1 | B3F-3100 | B3F-31XX |
| 1 | U$2 | 282837-6 | 282837-6 |
| 1 | X4 | MSTBA4 | MSTBA4 |
| 1 |  | 0.1” 18x2 pin |  |
| 8 |  | 0.1” 2x3 pin |  |
| 5 |  | 0.1” 8x1 pin |  |
| 2 |  | 0.1” 6x1 pin |  |
| 6 |  | 0.1” 4x1 pin |  |
| 2 |  | 0.1” 24x1 female |  |
| 4 |  | 0.1” 8x1 female |  |
| 15 |  | 0.1” 2 pin jumper |  |


RAMPS 1.4 (RepRap Arduino Mega Pololu Shield)
reprap.org/wiki/RAMPS1.4
Copyright 2011 Johnny Russell
UltiMachine
GPL v3
MEGA Conn.
U1
ARDUINO_MEGA_SHIELD
X-STEP
X-DIR
Y-EN
Y-STEP
Y-DIR
Z-EN
THERM0
THERM1
THERM2
RESET
AM-VIN
VCC
GND
X-MAX
X-MIN
Y-MIN
Y-MAX
Z-MIN
Z-MAX
SDA
SCL
E0-EN
E0-STEP
E0-DIR
E1-EN
E1-STEP
E1-DIR
Z-STEP
Z-DIR
X-EN
MISO
MOSI
SCK
LED
LED
1K
R12
LED1
Endstops
X-
X+
Y-
Y+
Z-
Z+
X-MIN
X-MAX
Y-MIN
Y-MAX
Z-MIN
Z-MAX
Thermistors
JP7
Thermistor 0
Thermistor 1
Thermistor 2
R7
R1
R11
4.7K
C5
C8
C1
10uF
I2C
R22
R21
4.7K
I2C
Heaters & Fans
STP55NF06L
Q1
Q2
Q3
D10
D9
D8
R13
R14
R15
10r
R9
R2
R4
100k
LED4
LED3
LED2
1.8K
R25
R24
R23
+12V
+12V2
P$1
P$2
P$3
P$4
P$5
P$6
Stepper Drivers
E0
E1
X
Y
Z
A4988 Carrier
JP2
JP3
JP4
JP5
JP6
E0-MOT
E1-MOT
X-MOT
Y-MOT
Z-MOT1
Z-MOT
C3
C4
C7
C9
C10
100uF
Power
VCC
PS-ON
D12
5V
X4-1
X4-2
X4-3
X4-4
MSTBA4
F2
MFR1100
MFR500
F1
D1
1N4004
D2
1N4004
C6
100uF
C2
100nF
12V-AUX
Reset
S1
RESET
Servos
SER1
SER2
SER3
SER4
D11
D6
D5
D4
+5V
AUX-1
A-OUT
SERIAL
A3
A4
AUX-2
U$10
M05X2
A5
A10
D44
A12
A9
D40
D42
A11
AUX-3
U$3
M04X2
MISO
SCK
D49
MOSI
D53
AUX-4
U$9
M18_
D16
D17
D23
D25
D27
D29
D31
D33
D35
D37
D39
D41
D43
D45
D47
D32
OSHW LOGO

# RAMPS 1.4 (RepRap Arduino MEGA Pololu Shield) GPL v3

reprap.org/wiki/RAMPS1.4

Reversing input power, and inserting stepper drivers incorrectly will destroy electronics.



12-35V Out

| | |
|---|---|
| D10 | − |
| | + |
| D9 | − |
| | + |
| D11 | − |
| | + |

12-35VDC In

| | |
|---|---|
| 11A | + |
| | − |
| 5A | + |
| | − |

SERVOS

| | | | |
|---|---|---|---|
| D11 | D6 | D5 | D4 |
| 5V | 5V | 5V | 5V |
| GND | GND | GND | GND |

AUX-1

| | | | |
|---|---|---|---|
| 5V | GND | D2 | D1 |
| 5V | GND | A3<br>D57 | A4<br>D58 |

AUX-2

| | | | | |
|---|---|---|---|---|
| GND | A9<br>D63 | D48 | D42 | A11<br>D65 |
| 5V | A5<br>D59 | A10<br>D64 | D44 | A12<br>D66 |

AUX-3 / SPI

| | | | |
|---|---|---|---|
| GND | SCK<br>D52 | MISO<br>D50 | 5V |
| NC | D53 | MOSI<br>D51 | D49 |

I2C

| |
|---|
| 21 |
| 20 |
| GND |
| 5V |

AUX-4

| |
|---|
| D16 |
| D17 |
| D23 |
| D25 |
| D27 |
| D29 |
| D31 |
| D33 |
| D35 |
| D37 |
| D38 |
| D41 |
| D43 |
| D45 |
| D47 |
| D32 |
| GND |
| 5V |

Copyright 2011 Johnny Russell - UltiMachine

PBC LINEAR
A PACIFIC BEARING CO.

# **Stepper Motor** NEMA 17

**This document describes mechanical and electrical specifications for PBC Linear stepper motors; including standard, hollow, and extended shaft variations.**

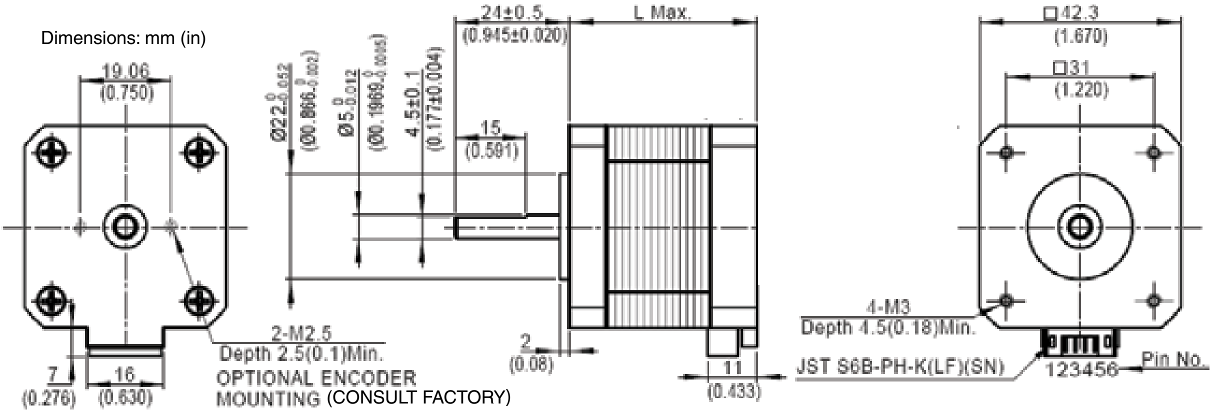
Standard shaft motor shown.

| | |
|---|---|
| Phases | 2 |
| Steps/Revolution | 200 |
| Step Accuracy | ±5% |
| Shaft Load | 20,000 Hours at 1000 RPM |
| Axial | 25 N (5.6 lbs.) Push |
| | 65 N (15 lbs.) Pull |
| Radial | 29 N (6.5 lbs.) At Flat Center |
| IP Rating | 40 |
| Approvals | RoHS |
| Operating Temp | -20° C to +40° C |
| Insulation Class | B, 130° C |
| Insulation Resistance | 100 MegOhms |

| Description | Length | Mounted Rated Current | Mounted Holding Torque | | Winding Ohms | mH | Detent Torque | | Rotor Inertia | | Motor Weight | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (Stack) | "L" Max | Amps | Nm Typ. | oz-in Typ. | ±10% @ 20°C | Typ. | mNm | oz-in | g cm2 | oz-in2 | kg | lbs |
| Single | 39.8 mm (1.57 in) | 2 | 0.48 | 68 | 1.04 | 2.2 | 15 | 2.1 | 57 | 0.31 | 0.28 | 0.62 |
| Double | 48.3 mm (1.90 in) | 2 | 0.63 | 89 | 1.3 | 2.9 | 25 | 3.5 | 82 | 0.45 | 0.36 | 0.79 |
| Triple | 62.8 mm (2.47 in) | 2 | 0.83 | 120 | 1.49 | 3.8 | 30 | 4.2 | 123 | 0.67 | 0.6 | 1.3 |

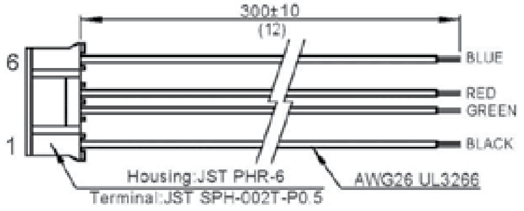
*All standard motors have plug connector. Consult factory for other options.

Dimensions: mm (in)



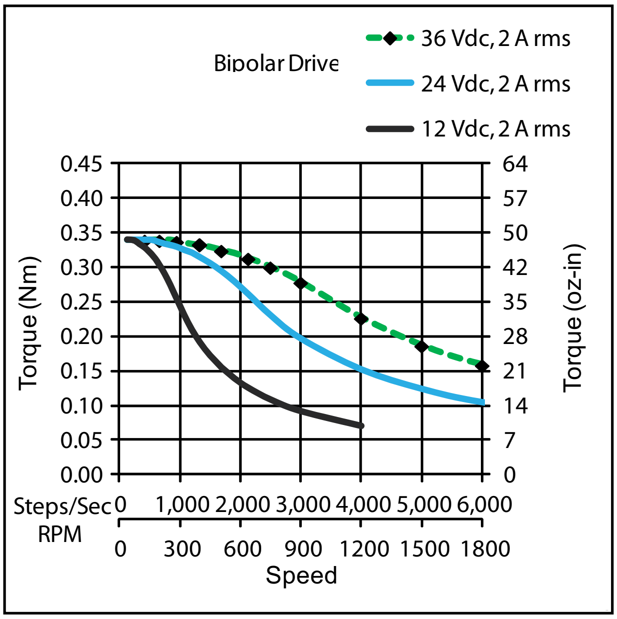
Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

Dimensions: mm (in)
4 Lead Connector, PBC Part#6200490
(Consult factory for optional motor connectors)

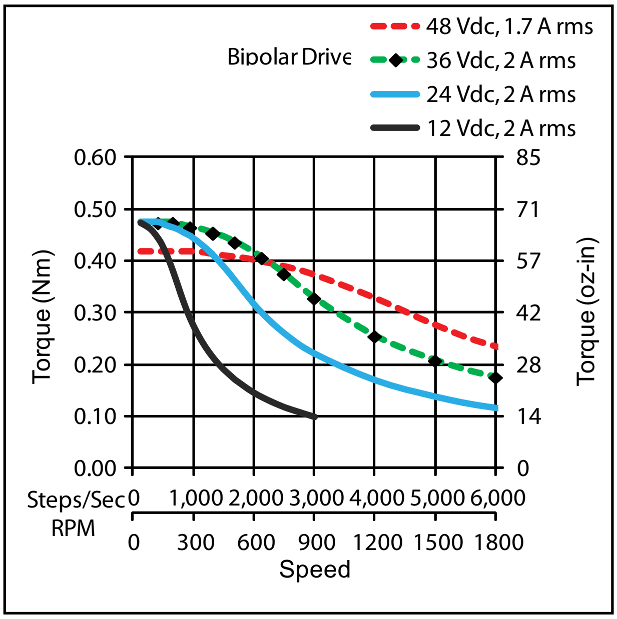


# NEMA 17 **Stepper Motor**

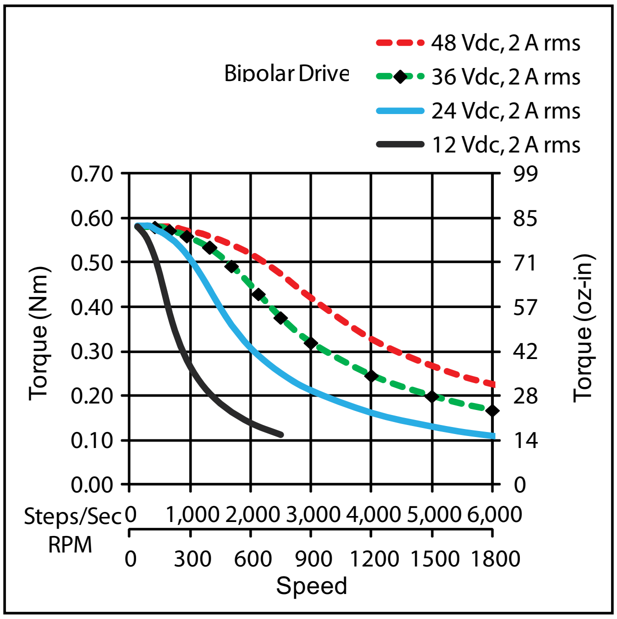
*Performance curves apply to continuous duty cycles.
Consult factory for intermittent cycles or other voltages.



Single Stack

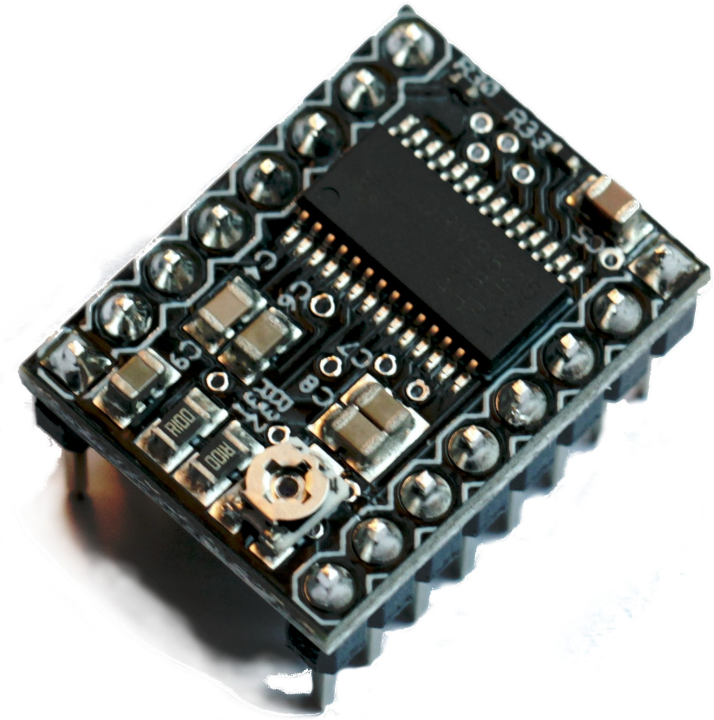


Double Stack



Triple Stack

# STEPSTICK DRV8825 v1.0 DATASHEET



| | |
|---|---|
| **Author** | Bart Meijer |
| **Date** | 21st of October 2013 |
| **Document version** | 1.0 |

ReprapWorld.com

## PRODUCT OVERVIEW

The stepstick DRV8825 is a breakout board for the Texas Instruments DRV8825 stepper motor controller. You can use this board to act as interface between your microcontroller and stepper motor. The DRV8825 is able to deliver up to 2.5A and can be controlled with a simple step/direction interface. The controller has a resolution of min. 1/32 step and protective features for over-current, short circuit and over-temperature. See the DRV8825 Datasheet for details on the DRV8825 controller.

The stepstick DRV8825 supersedes the stepstick A4988, which has been discontinued. The aim is for the stepstick DRV8825 to be a drop-in replacement for Stepstick A4988.

## SAFETY WARNINGS

Always disconnect the power source from the board before unplugging the stepper motor and/or adjusting the current. Failure to do so may result in permanent damage to the board and/or injuries due to high voltage spikes.

The stepper driver may get **HOT**, do not touch the device until it had a few minutes to cool down after operation.

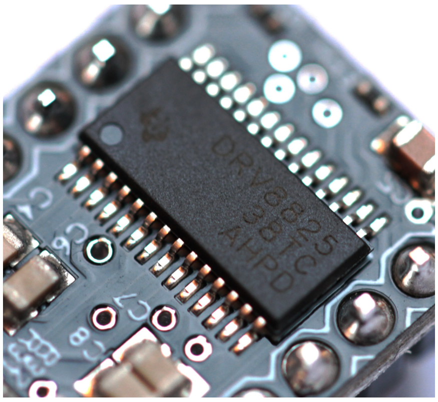
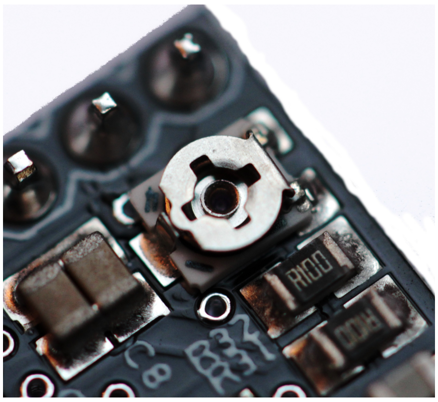
It is recommended to drive the stepper motor on current as low as possible to reduce power consumption and increase lifespan.

It is **NOT** recommended to turn the stepper motor while connected to the electronics. While turning the stepper motor, large voltages may be emitted through the VMOT pin, which can damage the electronics.

## TECHNICAL SPECIFICATION

| | |
|---|---|
| **Controller** | DRV8825 |
| **Operating Voltage (logic)** | 3-5.25V |
| **Operating Voltage (vmot)** | 12-24V |
| **Max current** | 2.5A |
| **Dimensions** | 20.4x15.6mm |

## MAJOR FEATURES

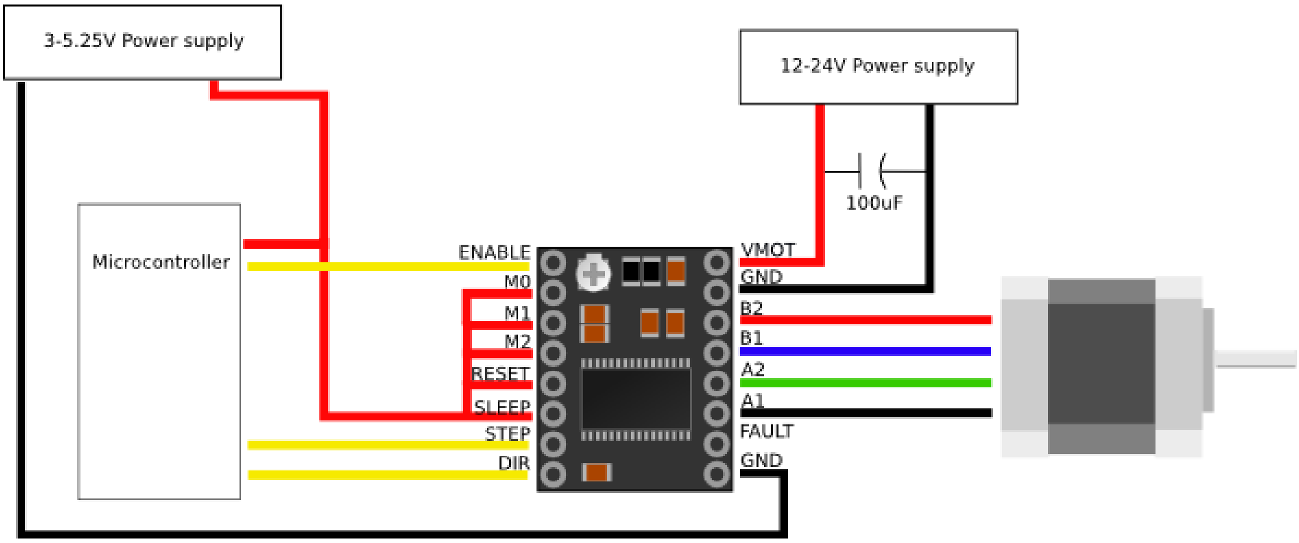
| | |
|---|---|
|  | **DRV8825**<br>Powerful DRV8825 with<br>• High current driver capable up to 2.5A<br>• Six different step resolutions: full-step, half-step, 1/4-step, 1/8-step, 1/16-step, and 1/32-step<br>• Protection against over-temperature and over-current<br>• No logic voltage required |
|  | **Adjustable current**<br>Using the trimpot on the board you can easily turn the current up or down. Turn left to lower the current, right to output a higher current. |

## OTHER FEATURES

- Four layer high quality PCB board
- Pre-soldered, no need to solder the headers

## EXAMPLE CIRCUIT

The following diagram lists the pins and show an example circuit.



| Name | Description |
|---|---|
| Enable | Enable/disable the stepper driver<br>HIGH – Disable<br>LOW – Enable * |
| M0 – M2 | Step resolution setting, see chapter 'step resolution configuration' |
| RESET | Enable/disable  the H-bridge output *<br>LOW – Disable *<br>HIGH – Enable |
| SLEEP | Enable/disable low-power sleep mode<br>LOW – Sleep *<br>HIGH – Active |
| STEP | LOW → HIGH, move one step |
| DIR | LOW / HIGH switches direction |
| VMOT | Motor power (12-24V) |
| GND | System ground |
| FAULT | LOW when the stepper driver is in fault condition. You can provide 5V on this pin for compatibility with stepstick A4988 |

* this is the default state when the pin is not connected

## STEP RESOLUTION CONFIGURATION

The DRV8825 had six step resolution modes, which can be configured using the M0-M2 pins on the stepstick DRV8825. The following table lists the step resolution settings:

| M0 | M1 | M2 | Resolution |
|---|---|---|---|
| Low | Low | Low | Full step |
| High | Low | Low | Half step |
| Low | High | Low | 1/4 step |
| High | High | Low | 1/8 step |
| Low | Low | High | 1/16 step |
| High | Low | High | 1/32 step |
| Low | High | High | 1/32 step |
| High | High | High | 1/32 step |

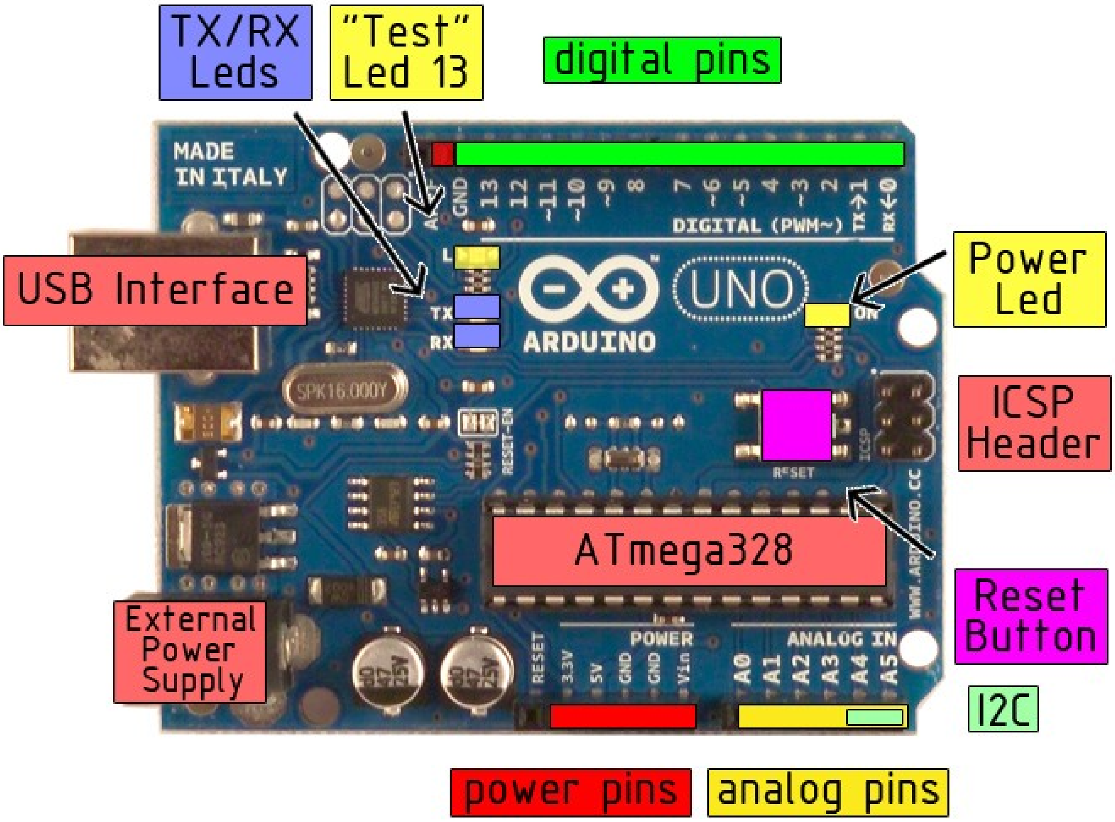
# Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

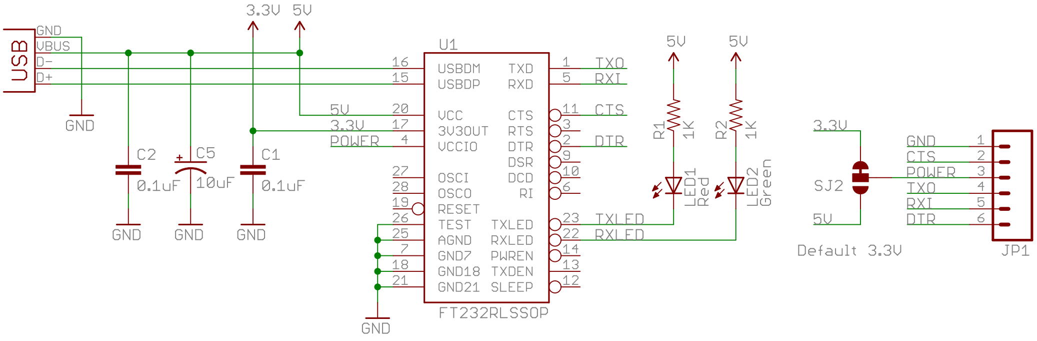
## Summary

| | |
|---|---|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB of which 0.5 KB used by bootloader |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |

## the board



RS radiospares RADIONICS ALLIED ELECTRONICS AN ELECTROCOMPONENTS COMPANY



TITLE: FTDI Basic-v22-3.3V

SFE SFE

Document Number:

REV:

Date: 11/14/2011 10:28:45 AM

Sheet: 1/1