Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Espen Brandtzæg Godø
Mari Ellinor Longva Haram
Ole-Andre Aurvåglid Ramstad

Improving the testing of the Mcon propulsion control system

Bachelor's thesis in Electrical Engineering - Automation and Robotics
Supervisor: Ottar L. Osen
Co-supervisor: Robin T. Bye
May 2022

NTNU
Norwegian University of
Science and Technology

Espen Brandtzæg Godø
Mari Ellinor Longva Haram
Ole-Andre Aurvåglid Ramstad

Improving the testing of the Mcon propulsion control system

Bachelor's thesis in Electrical Engineering - Automation and Robotics
Supervisor: Ottar L. Osen
Co-supervisor: Robin T. Bye
May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Kunnskap for en bedre verden

Improving the testing of the Mcon propulsion control system

Espen Brandtzæg Godø

Mari Ellinor Longva Haram

Ole-Andre Aurvåglid Ramstad

May 2022

BACHELOR THESIS

Department of ICT and Natural Sciences

Norwegian University of Science and Technology

Supervisor: Ottar L. Osen

Supervisor: Robin T. Bye

Preface

This bachelor thesis is written by three students from Electrical Engineering - Automation and Robotics at NTNU Ålesund. The project and thesis were carried out in the spring of 2022 and were done for Kongsberg Maritime. What motivated us to choose this task was the general interest in the maritime industry and the automation within it. This project gave us the opportunity to work with advanced control systems and develop solutions for real situations. The purpose of the project was to expand and improve on an already built solution. The project included tasks such as software development, mathematical model design, data processing, and testing.

We thank Kongsberg Maritime for giving us the opportunity to work on such an exciting task for our bachelor's project, and all the knowledge they have provided us with along the way. We also express our gratitude to the people who helped us during the project.

- Håkon Lunheim, Geir Olav Otterlei and Truls Antonsen from Kongsberg Maritime for their guidance and encouragement during the project.
- Stig Worren from Kongsberg Maritime for his technical assistance and guidance.
- Our advisors Ottar L. Osen and Robin T. Bye at NTNU, for their guidance with the project.
- Torgeir Sundet from Wago Support Norge for his expertise and assistance.
- Friends and family for help and support throughout the project period.

Summary and Conclusions

This project aimed to develop and improve a thruster simulator to test the Mcon propulsion control system using an open-source development platform.

The thruster simulator is built up of multiple parts, where each part simulates a function in the thruster. The implemented functions consisted amongst other things of a propeller model with adjustable propeller blade angle, thruster rotation, and logic to control pumps. The propeller model can simulate the response based on the propeller speed, the fluid speed, and the angle of the propeller blade. Two new thruster types were made from the functions and tested with a physical test system. A communication protocol was added to the system to gather the results. To further help the project's execution, several tools and programs were developed.

The results show that both thruster types were successfully implemented and tested. Compared to the previous thruster simulator, the functionality was extended with multiple thruster and propeller types. There was a reasonable response for the models considering the simplifications. This project uses an open-source platform, which causes limitations in the implementation of some features and complicates the project.

The final product can be used to test some of the functionality of the Mcon control system, but there are several areas for improvement.

Contents

List of Figures

List of Tables

Terminology

Azimuth Angular rotation of the thruster about its own axis

Binary compatible Two controller that can run the same binary program

Bow The front end of a ship

Containerised development All requirements for a program is segmented into a single unit

Instruction set Details how a processor handles operations and implements functions

Microarchitecture How an instruction set is implemented in a processor

Open Source Source code freely available for anyone to study and use

Pitch Angle between the propeller blade and the plane of rotation of the propeller [1]

Port Left side of a ship when standing on a ship and facing the front end

Runtime When a computer is executing a program

Starboard Right side of a ship when standing on a ship and facing the front end

Stern the rear end of a ship

Thruster Device that produces thrust by moving a quantity of air or water

Toolchain Software tool used to execute advanced development tasks

Notation

K_d Derivative term of a PID controller

Kg System International unit for Kilogram

K_i Integral term of a PID controller

K_p Proportional term of a PID controller

v_b Counter-electromotive force [V]

K_b Counter-electromotive force constant $[V/m/s]$

T_m Motor torque $[Nm]$

K_t Torque constant

J_m Total motor inertia $[kg-m^2]$

J_l Load inertia $[kgm^2]$

D_m Motor dampening $[Ns/m]$

Θ_m Motor rotation $[m/s]$

ω_m Angular velocity of the motor $[rad/s]$

N Gear ratio

Q Hydrodynamic torque $[Nm]$

R Propeller radius $[rad]$

U_a Fluid velocity $[m^3/s]$

V Total relative fluid velocity $m^3/s]$

a_e Angle of attack $[rad]$

θ Relative flow angle $[rad]$

T Axial thrust $[N]$

Abbreviations

AO Analog out

AI Analog in

CCW Counterclockwise

CLI Command-Line Interface

CPP Controllable Pitch Propeller;

CPU Central Processing Unit;

CW Clockwise

DI Digital in

DO Digital out

DP Dynamic Positioning;

FAT Factory acceptance test; test procedure to certify equipment before being delivered from factory

FPP Fixed Pitch Propeller

GUI Graphical User Interface

IDE Integrated Development Environment

IEC International Electrotechnical Commission

IEEE Institute of Electronical and Electronic Engineers

KVL Kirchhoff's Voltage Law

PID Proportional Integral Derivative Controller

PLC Programmable Logic Controller

PM Permanent Magnet

POU Program Organisational Unit

PWM Pulse Width Modulation;

RPM Rotations Per Minute

SDK Software Development Kit

SOC System On a Chip;

TCP Transmission Control Protocol;

UDP User Datagram Protocol;

Chapter 1

Introduction

1.1 Background

The project was presented by Kongsberg Maritime, a company in marine technology. The project goal was to improve and expand the product of a Bachelor's project which was completed a year prior. The goal of the previous project was to improve and streamline the testing of the Mcon propulsion control system at their Longva facility [2]. The test and verification of the Mcon propulsion control system is a tedious and time-consuming task when using a complete physical setup. A thruster simulator was created to replace physical test equipment with software to reduce testing time. Then parts of the Factory Acceptance Test (FAT) testing were semi-automated by a Java program.

The main objective of this thesis is to further develop this solution by adding more functionalities and removing some limitations.

1.2 Problem Formulation

The project advisors in Kongsberg Maritime had some ideas they would like to see implemented in the project, but were also open to suggestions from the project group. Together, the advisors and the project group decided on three main activities for the project. A fourth activity was established as a stretch goal to expand the project, given enough time in the project.

The main activities were porting the thruster simulator to open source software, expanding the types of thrusters available for simulation in the thruster simulator, and improving the dynamic model designed in the previous bachelor thesis. The stretch goal was to implement visualisation to the thruster simulator.

Writing the bachelor thesis should be considered the fifth activity. The activities are listed below.

- Activity 1: Port thruster simulator to Eclipse 4diac
- Activity 2: Expand with additional thruster models
- Activity 3: Improve existing dynamic models
- Activity 4: Add visualisation (Stretch)
- Activity 5: Thesis

1.3 Limitations

In this project, some limitations were established to limit the scope of the project and to maintain compatibility with the previous thruster simulator. The Kongsberg advisors requested the use of 4diac to develop the thruster simulator. To limit the workload of the project, only two new thruster types should be added. Since the semi-automated FAT from the earlier thesis uses OPC UA, it was deemed necessary to incorporate OPC UA communication to maintain backward compatibility. The summarised limitations are listed below.

- Use 4diac as the open source software
- Add only two types of thrusters
- Incorporate OPC UA

1.4 Approach

During the pre-project it was decided to divide the project into five main activities. The first activity consisted of porting the thruster simulator from the B&R Programmable Logic Controller

(PLC) to Wago PLC running 4diac. This involved researching the 4diac software, porting the old system, and verifying if the semi-automated FAT still worked. The second activity consisted of expanding the thruster selection in the simulator. During this activity, different thrusters were researched and provided the foundation for mathematical models.

The third activity was to improve on the thruster model made by the previous bachelor project. The fourth activity was a stretch goal to include visualisation in the project. The visualisation idea was to implement it into Morild Interactive, a ship and bridge simulator used for training. The fifth and final activity was to write and complete the thesis.

1.5 Structure of the Report

The remainder of the report is structured as follows.

Chapter 2 - Theoretical basis: Contains an introduction to the theoretical background required for this project.

Chapter 3 - Materials: Contains a description of the materials and software that were used throughout the project.

Chapter 4 - Method: Contains a description of the methodology that was considered throughout the project.

Chapter 5 - Result: Contains a description of the finished project and the results obtained.

Chapter 6 - Discussion: A summary of the results, the solutions used, why they were used, and potential improvements

Chapter 7 - Conclusions: This chapter presents an overall conclusion of the project and the solution presented.

Chapter 2

Theoretical basis

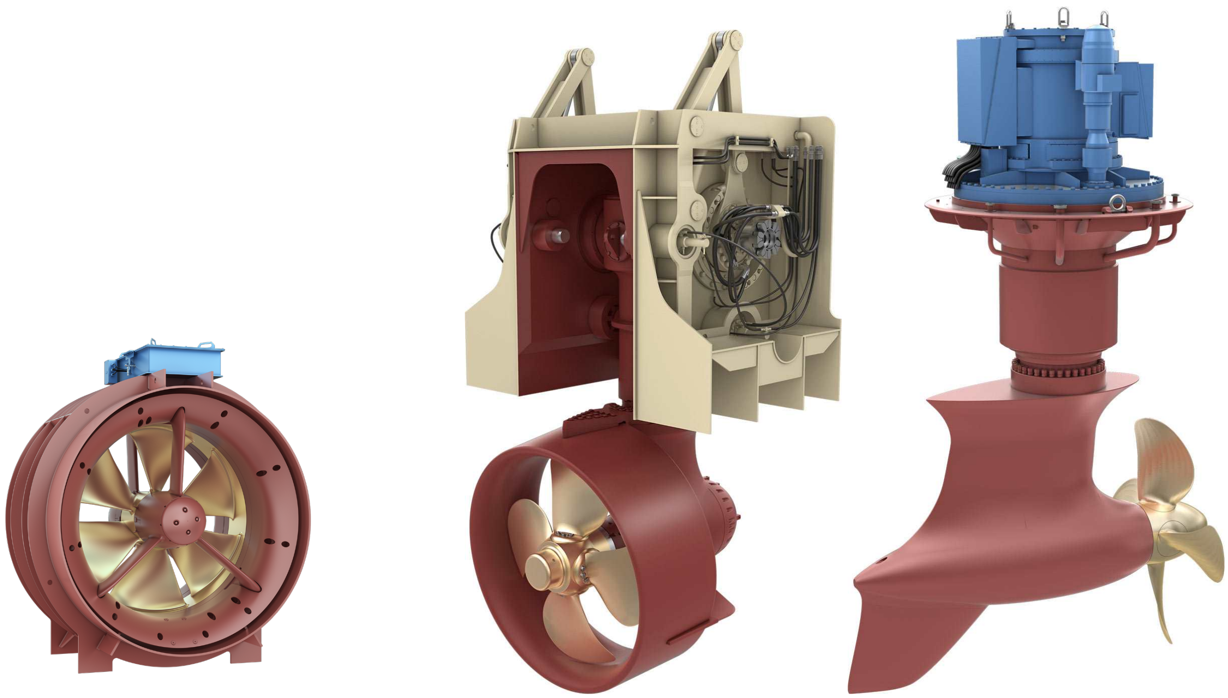
This chapter will provide the necessary explanations of the theoretical basis that is the foundation for this project. Multiple subjects will be presented, such as thruster types, modelling concepts, programming standards, and software.

2.1 Thrusters

A wide range of thrusters are available for a variety of applications. This subsection provides an overview of the three thruster types involved in this project and some of their key features.

2.1.1 Tunnel Thruster

Tunnel thrusters such as in Figure 2.1a are used for side propulsion to make a ship more manoeuvrable [3]. They are mounted in the hull of the ship, usually in the front end (bow) or rear end (stern), which means that the direction of the propeller cannot be changed. Thus it can only give thrust in the port (left side when facing the bow) or starboard (right side when facing the bow) direction [3]. This helps to turn the ship without forward propulsion, which is especially useful during docking [3]. Tunnel thrusters are usually delivered with a Fixed Pitch Propeller (FPP).



(a) (b) (c)

Figure 2.1: The different types of thrusters used. (a) Tunnel thruster PM 2000 [4]. (b) Swing-up azimuthing thruster [5]. (c) Azipull thruster [6].

2.1.2 Azipull Thruster

The Azipull thruster as seen in Figure 2.1c is mounted on the underside of the hull. It pulls itself through the water, in contrast to most thrusters and propellers, which push water [6]. It is of the azimuth type, which means that the thrust direction can rotate 360°[6]. The Azipull thruster can be delivered as a controllable pitch propeller (CPP), which means that the pitch can be adjusted [6].

2.1.3 Swing-up Azimuthing Thruster

The Swing-up Azimuthing thruster, as seen in Figure 2.1b, has similar features to the Azipull thruster, except it pushes the water instead of pulling [5]. It has rotatable thrust direction and can be delivered as a CPP [5]. In addition, it can swing up into the hull when not in use. When swung up, it does not protrude below the hull and can therefore be a useful option for shallow water operations [5].

2.2 Modelling

This section gives an overview of the modelling theory used in this project. This includes the different dynamic models as well as an RC circuit.

2.2.1 RC Circuit

An RC circuit with a voltage divider such as in Figure 2.2 can be used to convert a Power Width Modulation (PWM) signal to an analogue signal.

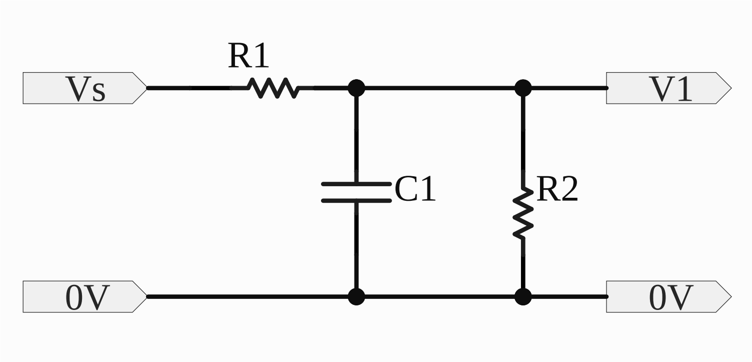


Figure 2.2: Schematic of an RC circuit with a voltage divider.

The current in the node connecting C_1, R_1 and R_2 is defined through KCL:

$$i_{R1}(t) = i_{C1}(t) + i_{R2}(t) \tag{2.1}$$

If the current going to V_1 is 0. From Ohms law, i_{R_1} and i_{R_2} can be defined based on the differential voltage over the resistor and the resistance. The equation that relates the current through a capacitor with the change in voltage, which can be found in multiple engineering textbooks [7]–[9], is defined as:

$$i_{R1}(t) = \frac{V_s(t) - V_1(t)}{R_1}; \quad i_{C1}(t) = C\frac{dV_1(t)}{dt}; \quad i_{R2}(t) = \frac{V_1(t)}{R_2} \tag{2.2}$$

By substituting 2.2 into 2.1 we get the following first-order differential equation:

$$\frac{V_s(t) - V_1(t)}{R_1} = C\frac{dV_1(t)}{dt} + \frac{V_1(t)}{R_2} \tag{2.3}$$

Furthermore, this equation can be Laplace transformed to 2.4, and with basic arithmetic, the transfer function can be 2.7. This transfer function can also be written in standard form as 2.8.

$$\frac{V_s(s) - V_1(s)}{R_1} = CsV_1(s) + \frac{V_1(s)}{R_2} \tag{2.4}$$

$$\frac{V_s(s)}{R_1} = \frac{V_1(s)}{R_1} + CsV_1(s) + \frac{V_1(s)}{R_2} \tag{2.5}$$

$$\frac{V_s(s)}{R_1} = \left(\frac{1}{R_1} + \frac{1}{R_2} + Cs\right) V_1(s) \tag{2.6}$$

$$\frac{V_1(s)}{V_s(s)} = \frac{1}{1 + \frac{R_1}{R_2} + R_1 Cs} \tag{2.7}$$

$$\frac{V_1(s)}{V_s(s)} = \frac{\frac{R_2}{R_1+R_2}}{\frac{R_1 R_2}{R_1+R_2} Cs + 1} \tag{2.8}$$

2.2.2 DC motor dynamics

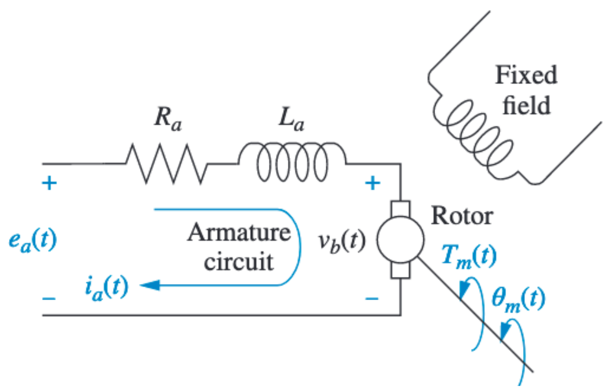


Figure 2.3: Principal diagram of a DC-motor. Copied from [7].

A simplified DC motor can be modelled from the diagram shown in Figure 2.3.

When a conductor moves in a magnetic field, a voltage called the back-emf is generated that is proportional to the velocity of the conductor [7]. The back-emf $v_b(t)$ that occurs in the motor is directly proportional to the angular velocity of the rotor [7].

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \tag{2.9}$$

By using the KVL on the circuit the following equation is found:

$$e_a(t) = v_{R_a}(t) + v_{L_a}(t) + v_b(t) \tag{2.10}$$

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \frac{d\theta_m(t)}{dt} \tag{2.11}$$

The torque $T_m(t)$ of the rotor is proportional to the current through the rotor:

$$T_m(t) = K_t i_a(t) \tag{2.12}$$

Where K_t is the torque constant and is dependent on the motor and magnetic field characteristics [7]. From 2.12 it can be derived that $i_a(t)$ can be substituted with $\frac{T_m(t)}{K_t}$ in 2.11 to achieve the equation 2.13. If we assume that L_a is small compared to R_a the equation can be simplified to 2.14 [7].

$$e_a(t) = \frac{\left(R_a + L_a \frac{d}{dt}\right)}{K_t} T_m(t) + K_b \frac{d\theta_m(t)}{dt} \tag{2.13}$$

$$e_a(t) = \frac{R_a}{K_t} T_m(t) + K_b \frac{d\theta_m(t)}{dt} \tag{2.14}$$

The torque can further be modelled with respect to the position, with the inertia J_m and dampening D_m experienced by the motor to achieve 2.15 [7]. By substituting this equation into 2.14 to get the relation between the input voltage and the position, see 2.16 and 2.17.

$$T_m(t) = J_m \frac{d^2\theta_m(t)}{dt^2} + D_m \frac{d\theta_m(t)}{dt} \tag{2.15}$$

$$e_a(t) = \frac{R_a}{K_t}\left(J_m \frac{d^2\theta_m(t)}{dt^2} + D_m \frac{d\theta_m(t)}{dt}\right) + K_b \frac{d\theta_m(t)}{dt} \tag{2.16}$$

$$e_a(t) = \frac{R_a J_m}{K_t} \frac{d^2\theta_m(t)}{dt^2} + \left(\frac{R_a D_m}{K_t} + K_b\right) \frac{d\theta_m(t)}{dt} \tag{2.17}$$

The transfer function for the model can be found by using the laplace transformation on 2.17 to get 2.18 and rearranging it to 2.19.

$$E_a(s) = \frac{R_a J_m}{K_t} s^2 \Theta_m(s) + \left(\frac{R_a D_m}{K_t} + K_b \right) s \Theta_m(s) \tag{2.18}$$

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{\frac{K_t}{R_a J_m}}{s\left(s + \frac{D_m}{J_m} + \frac{K_t K_b}{R_a J_m}\right)} \tag{2.19}$$

By definition, the derivative of the position is the velocity. Thus, if we define the rotational velocity of the rotor as shown in 2.20 and substitute it into 2.17, a mathematical model can be derived to 2.21.

$$\omega_m(t) = \frac{d\theta}{dt} \tag{2.20}$$

$$e_a(t) = \frac{R_a J_m}{K_t} \frac{d\omega_m(t)}{dt} + \left(\frac{R_a D_m}{K_t} + K_b \right) \omega_m(t) \tag{2.21}$$

$$\frac{d\omega_m(t)}{dt} = -\left(\frac{D_m}{J_m} + K_b \right) \omega_m(t) + \frac{K_t}{R_a J_m} e_a(t) \tag{2.22}$$

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{\frac{K_t}{R_a J_m}}{s + \frac{D_m}{J_m} + \frac{K_t K_b}{R_a J_m}} \tag{2.23}$$

J_m is the total moment of inertia that affects the motor. If a mechanical load is connected to the motor through a gear, the total inertia can be calculated with:

$$J_m = J_a + J_L \frac{1}{N^2} \tag{2.24}$$

where J_a is the inertia of the motor, J_L is the inertia of the load and the gear ratio is N. Similarly, the damping D_m, which can also be calculated, is the total damping:

$$D_m = D_a + D_L \frac{1}{N^2} \tag{2.25}$$

2.2.3 Propeller dynamics

In 1995 Healy et al. proposed a new four quadrant thruster model [10] as seen in Figure 2.4. This model uses an estimate of the lift and drag combined with an estimate of water flow to estimate all four quadrants [10]. The paper follows the development of the model in [11]–[15] where it is

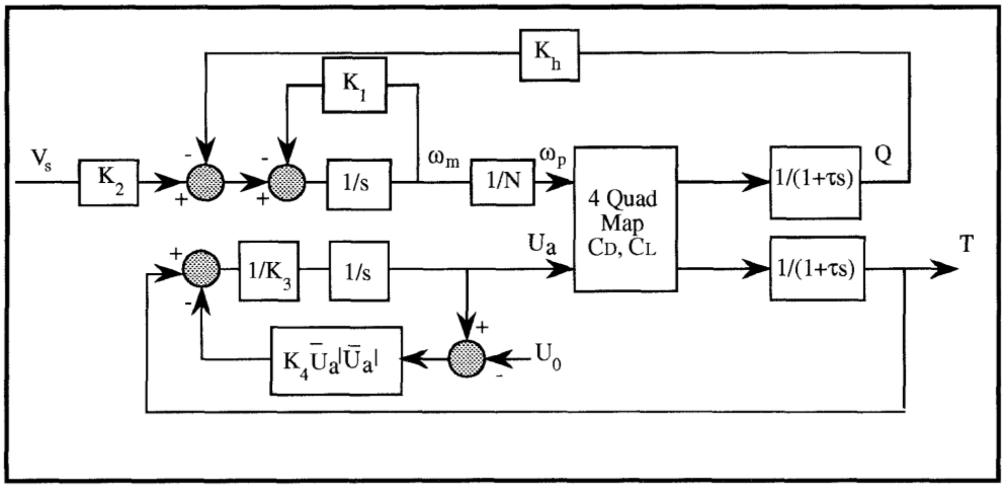


Figure 2.4: Block diagram of motor shaft speed and water velocity from Healey et al. [10].

described and tested in more detail. The model is summarised below based on the description given in [10] and [11]. The model uses the standard model for a DC motor similar to the one described in 2.2.2. However, it also incorporates one additional term in the differential equation 2.22 to include how the hydrodynamic torque Q acts on the change in velocity of the motor shaft [11] and gets 2.26.

$$\frac{d\omega_m(t)}{dt} = -\left(\frac{D_m}{J_m} + K_b\right)\omega_m(t) + \frac{K_t}{R_a J_m} e_a(t) - \frac{Q}{J_m N} \tag{2.26}$$

Where K_2 is the total inertia seen from the motor. The model assumes that the motor is connected to a propeller with the gear ratio N [10]. Further, the velocity of the fluid is relative to the tangential speed of the propeller measured at one radial position[10]. In the model, they used $0.7R$, where R is the radius of the propeller [10]. Therefore, this relation is formulated as follows:

$$U_p = \frac{0.7R}{N}\omega_m \tag{2.27}$$

The velocity of the incoming fluid is denoted as U_a, so the total relative velocity squared acting on the propeller can be described with [10]:

$$V^2 = U_p^2 + U_a^2 \tag{2.28}$$

The model for the lift and drag forces that affect the propeller is given as:

$$\text{Lift} = 0.5\rho V^2 A C_{Lmax} \sin(2\alpha_e) \tag{2.29}$$

$$\text{Drag} = 0.5\rho V^2 A C_{Dmax} \sin(2\alpha_e) \tag{2.30}$$

Where there are only two disposable parameters, C_{Lmax} and C_{Dmax}. This model uses the angle of attack α_e of the fluid described as 2.31 [11]. The relative flow angle θ describes which quadrant the model is using based on the fluid velocity in the propeller U_p and the velocity of the incoming fluid U_a, see 2.32 [11]. For example if a vessel is moving forward and the thruster is used to increase the vessel speed, then both the velocity of the fluid in the propeller and the incoming fluid is positive. This means the model is operated in the first quadrant. As opposed to if the vessel is moving forward and the thruster is used to decrease the vessel speed, the incoming fluid is positive, whereas the velocity of the fluid in the propeller is negative. Therefore, the model operates in the second quadrant.

$$\alpha_e = (\frac{\pi}{2} - \text{pitch}) - \theta \tag{2.31}$$

$$\theta = \arctan\left(\frac{U_a}{U_p}\right) \tag{2.32}$$

From the lift, drag and the relative flow angle the axial thrust T and hydrodynamic torque Q can be described by [10]:

$$T = \text{Lift} \cdot \cos\theta - \text{Drag} \cdot \sin\theta \tag{2.33}$$

$$Q = 0.7R\left(\text{Lift} \cdot \sin\theta + \text{Drag} \cdot \cos\theta\right) \tag{2.34}$$

The derivation of the hydrodynamic model is omitted. However, it describes the relationship between the axial thrust T and the incoming fluid velocity U_a to be [10]:

$$T = \left(\rho A L\gamma\right) \frac{dU_a}{dt} + \left(\rho A \Delta\beta\right) \overline{U_a} |\overline{U_a}| \tag{2.35}$$

By denoting K_3 K_4 and $\overline{U_a}$ to be [10]:

$$K_3 = \rho AL\gamma; \quad K_4 = \rho A\Delta\beta; \quad \overline{U_a} = U_a - U_0 \tag{2.36}$$

The following differential equation describes the incoming fluid velocity:

$$\frac{dU_a}{dt} = -\frac{K_4}{K_3}\overline{U_a}|\overline{U_a}| + \frac{1}{K_3}T \tag{2.37}$$

To summarise, the model can be described by the two first-order nonlinear equations:

$$\frac{d\omega_m(t)}{dt} = -\left(\frac{D_m}{J_m} + K_b\right)\omega_m(t) + \frac{K_t}{R_a J_m}e_a(t) - \frac{Q}{J_m N} \tag{2.38}$$

$$\frac{dU_a}{dt} = -\frac{K_4}{K_3}\overline{U_a}|\overline{U_a}| + \frac{1}{K_3}T \tag{2.39}$$

2.3 IEC Standard

IEC stands for International Electrotechnical Commission and is an international standards organisation responsible for a variety of electrotechnical standards. This is to establish a common ground and some basic rules and instructions that everyone should follow within their respective fields [16].

2.3.1 IEC 61131-3

The 61131-3 standard is one of the most widely used standards within the industrial industry, and the standard defines the syntax and meaning of the programming languages available in the standard [17]. Within the standard, there are three graphical languages and one textual language. The graphical languages are ladder diagram, function block diagram and sequential flow diagram, and the textual language is structured text [17], [18].

2.3.2 IEC 61499

The 61499 standard defines the architecture of distributed systems. The standard aims to make the architecture more usable for future changes to industrial automation [19]. There are many differences between the standards 61499 and 61131-3. Some of the key differences include the type of system it is designed for, how the system is executed and the use of global variables [19]. Since the distributed system incorporates several subsystems which may need communication between them, the need for explicitly defined global variables was considered unnecessary, so in the 61499 standard, global variables have been removed [19]. The 61131-3 standard is designed for centralised systems, whereas the 61499 standard is designed for distributed systems. [19] How the two standards execute their programmes differs because the 61131 standard runs the system until completion on a cyclic scheme, whereas the 61499 standard runs until completion upon an event trigger [19].

2.4 Software

2.4.1 4diac

Eclipse 4diac is an open source PLC framework intended for distributed industrial systems based on IEC 61499 standard [20]. 4diac is split up into four parts; IDE, Forte, Lib and Sys. 4diac has its own integrated development environment and a runtime environment. The IDE is called 4diac IDE and contains a multitude of features. Some of the functions that the IDE contains are: the system explorer, which allows the user to manage their applications; the applications manager, which allows the modelling of function block networks; and the type editor, which gives the user the possibility to create different kinds of basic and composite function blocks [20]. 4diac's runtime environment is named Forte. Forte is a small and portable runtime environment made in C++ aimed to be used in small control devices for both 16 and 32 Bit [20]. Forte is currently documented to support only a handful of different controllers, such as Lego Mindstorm, Raspberry Pi, Bosch Rexroth PLC, and a few more [20].

2.4.2 Docker

Docker is an open platform for developing, shipping and designing applications using containerised development [21]. Docker consists of three major parts; the image, the container and the dockerfile. A docker image is a form of template for building docker images. Images includes the code, runtime, system tools, libraries and settings needed to build a container [22]. A container is a standalone instance of the image, which you can freely manipulate after its been instanced [21], [23].

The dockerfile is a text document containing all the possible commands which can be read by the image while assembling[22]. This enables the user to both customise and automate the assembly process. Docker is somewhat similar to a virtual machine in that they both simulate parts of the machine, but they simulate different parts. Virtual machines virtualises the physical system. Meanwhile, docker virtualises the operating system [23].

2.5 Communication

2.5.1 OPC UA

OPC is a basic communication interface between industrial automation devices and is the standard interface for general communication [24]. OPC UA builds on the original OPC standard and enables more complex communication and functions. Some of the functions are subscriptions, event notifications, access to server-based methods, and assigning read/write permissions [24]. Another feature added with the UA version is the address space; all data are stored in a hierarchy with their own node ID consisting of a name space and an index [24].

2.5.2 SSH

SSH stands for Secure Socket Shell and is a network protocol used to remotely access computer systems securely [25]. It features strong encryption and authentication with passwords and keys to ensure secure data transmission [25]. The primary uses of SSH are to access remote systems such as servers or controllers, where they can access the terminal on the connected device and perform the desired action such as remote commands or file transfer [25].

2.5.3 CANopen

CANopen is a high-level CAN-based communication protocol designed for embedded networking applications that are often used in automation [26]. It is made to simplify the development process of applications containing CAN without having to deal with hardware-specific details [26].

Devices that run CANopen consist of three logical parts; the protocol stack, the application software, and the object dictionary [27]. The stack handles communication, and the application software control the internal functionality and interfaces [27]. The dictionary contains all the information necessary for communication and applications [27].

Chapter 3

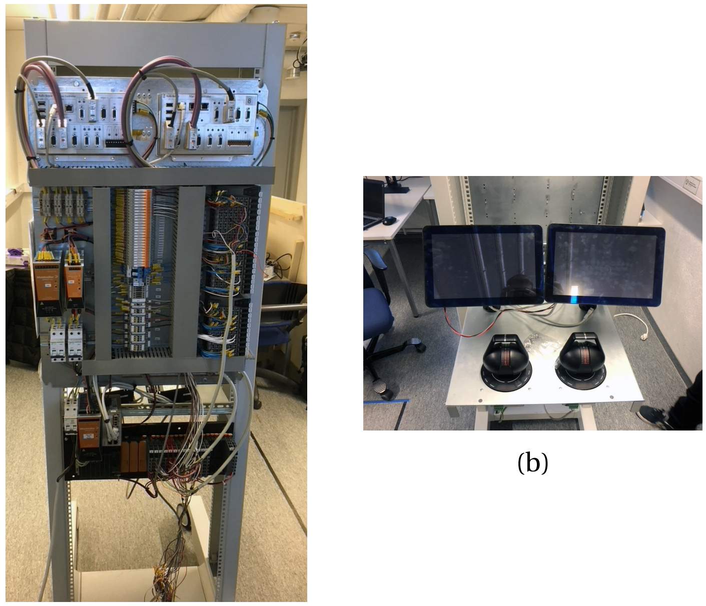
Materials

This chapter describes all materials and tools used to complete this project. This includes both existing and added materials, as well as software and external libraries for programming.

3.1 Mcon test rack

The thruster simulator used in this project was built in a previous bachelor thesis [2]. The thruster was delievered to as seen in Figure 3.1 The purpose of this simulator is to replace the need for access to the physical thruster and to make it easier to test the Mcon control system. The test rack can be divided into; interactive parts, logic parts, and power distribution. The interactive parts consist of two touch panel PCs, two indicators, and two motorised levers. The GUI on the panel PCs sends data to the Mcon and displays data from Mcon. Some of the data are also displayed on the indicators. The levers input data to Mcon and feature a digital display and motorised feedback on the lever not in command.

Power distributions consist of 24V power supplies, terminal blocks, and fuses. Finally, the logic part is built up by a B&R CPU, relays, two marine controllers, and two bus controllers. The marine controllers contain the Mcon system and communicate with the bus controllers using CANopen, which receives inputs from the thruster simulator running on the B&R CPU.



(a) (b)

Figure 3.1: How the Mcon test rack looked when received by the group.(a) Rear of the rack. (b) Front of the rack.

3.2 Added materials

This section will contain an overview of all materials added to the project to obtain the final product. This includes terminal blocks and PLC equipment.

3.2.1 Terminal blocks

Table 3.1 shows the terminal blocks with accessories that were added to the Mcon test jig.

Table 3.1: Material added

Parts name	**Wago item number**	**Quantity**
Terminal blocks	2102-5301	13
Jumpers	2002-410	2
End stop	249-116	2
End and intermediate plate	2002-1491	2

3.2.2 Wago I/O system

The components of the Wago I/O system added to the test rig are listed below.

- Wago CPU 750-8214
- Wago I/O modules, see Table 3.2

Table 3.2: Overview of added I/O

Item number	Quantity	Type	Name
750-513	8	DO	2-channel relay output; make contact
750-517	2	DO	4-channel relay output; changeover
750-430	3	DI	8-channel digital input
750-554	2	AO	2-channel analog output 4-20mA
750-556	2	AO	2-channel analog output ∓ 10V
750-455	2	AI	4-channel analog input 4-20 mA
750-459	2	AI	2-channel analog input 0-10V
750-626	1	Filter	24V Power supply filter (Surge)
750-600	1	End	End module

3.3 Tools and other materials

Table 3.3 shows the tools and materials used in the project.

Table 3.3: Tools and other materials used

Name	Quantity	Type	Serial number
Resistor	2	10kΩ	
Resistor	1	3.9kΩ	
Resistor	1	1kΩ	
Resistor	1	5.1kΩ	
Capacitor	2	2.2μ F	
Raspberry Pi	1	2B	
Breadboard	1		
Op-amp	1	lm358n	
Multimeter	2	Instrutek Mamy-60	VA140523527/VA120501689
70 MHz Digital oscilloscope	1	HMO722	018802613
25Mhz Arbitrary generator	2	HMF2525	0185339263/018320498
Programmable power supply	1	HMP2030	018802613

3.4 Software

Table 3.4 gives an overview of the various software used in the project. Table 3.5 shows the external libraries used in Python. The packages used in Ubuntu are listed in 3.6.

Table 3.4: Overview of the software used in the project

Software	**Version**	**Application**
Altium Designer	22.4.2	Used to make circuit schematic
App.diagrams.net		Online chartmaker
B&R Automation Studio	4.2	IDE to develop applications on B&R hardware
Eclipse 4diac IDE	2.0.1	IDE to develop applications which follows the IEC61499 standard
Eclipse Forte	2.0.1	PLC runtime
Cmake	3.23.0-rc3	Open source software used to build and test applications
Docker	20.10.14	Development software for containerised applications
Github		Online software development platform
IntelliJ IDEA Ultimate	2021.3.3	IDE for developing Java applications
MATLAB	R2021b	Computing environment for developing mathematical expressions and models
Microsoft Excel	2203	Spreadsheets
Microsoft Teams	1.5.00.8070	Communication and filesharing
Microsoft Visual Studio	1.66.0	Code editor with features such as build and compiling
Microsoft Visual Studio Code	17.1.1	General purpose IDE
Multisim Live		Online circuit simulation tool made by National Instruments
OPC UA Client	1.4	Software for connecting and interacting with OPC UA servers.
OPC UA Server Simulator	1.2	Host and customize OPC UA servers.
Overleaf		Web-based latex editor
Ptxdist	2022.05.0	Build and distribute images for embedded linux systems
Pycharm	2021.3.3	Python IDE
Simulink	R2021b	Diagram environment used for models and simulations
UAExpert	1.6.2	Software for connecting and interacting with OPC UA
Vmware	1.6.2.2	Virtual machines
Wireshark	3.6.3	Network protocol analyser.
PuTTY	0.76	SSH client

Table 3.5: External libraries used in Python and 4diac.

Python Library	**Version**	**Application**
Asyncio	3.10.4	Library to enable concurrent code in python
Asyncua	0.9.92	Combination of asynchronous execution and OPC UA interaction
Matplotlib	3.5.1	Library for making graphs and animations
Numpy	1.22	Enable advanced mathematics in python
Time	3.7.4	Advanced time and date commands
4diac Library	**Version**	**Application**
open62541	v1.0	Library to enable OPC UA in 4diac
mbedtls	2.7.1	Library to enable encryption of OPC communication

Table 3.6: Packages used in ubuntu

Software	**Version**	**Application**
autoconf	2.69-11	Automatic script configuration
autopoint	0.19.8.1-6ubuntu0.3	Tools for translating programming languages
bison	2:3.0.4.dfsg-1build1	General purpose parser
dialog	1.3-20171209-1	Displays dialog boxes from shell scripts
doxygen	1.8.13-10	Documentation for a set of programming languages
flex	2.6.4-6	Text analyser for language recognition
g++	4:7.4.0-1ubuntu2.3	C++ compiler
gawk	1:4.1.4	Pattern scanner and processing language
git	1:2.17.1-1ubuntu0.11	Git functionality
git-lfs	2.3.4-1	Git functionality for larger files
libc6-dev	2.27-3ubuntu1.5	Development libraries and headers
libncurses5-dev	6.1-1ubuntu1	Libraries for terminal handling
libtool	2.4.6-2	Generic library support script
lzop	1.03-4	Fast compression program
python-dev	2.7.15~rc1-1	Header files and library for python
python3-setuptools.	39.0.1-2	Support for installing and building modules in python
texinfo	6.5.0.dfsg.1-2	Documentation for on-line info and outputs
xmlstarlet	1.6.1-2	XML toolkit
xsltproc	1.1.29-5ubuntu0.2	Tool for applying stylesheets to XML documents

Chapter 4

Method

This chapter provides in-depth explanations of the methods used to solve the project's different parts. First, the project approach is described, focusing on how the project was planned and executed. Next, the setup and workflow of the project is described. This includes a detailed description of how the Forte runtime is built, and the workflow used to develop the system. This is followed by the description of the method used for modelling and programming. Finally, the complete system overview is described. This details how the system was set up and how the different parts were implemented.

4.1 Project approach

The group consisted of three students who shared the roles of project leader and secretary in rotation. It was decided to do it this way so that each member had the opportunity to have additional responsibilities. By circulating the roles, each member spends equal amounts of time on organisational tasks. The project leader's tasks were to plan and lead meetings, write meeting agendas, update the Gantt chart, and delegate tasks. The tasks for the secretary were to write and publish meeting minutes, and to be responsible for the progress report. All the project group members were responsible for completing assigned tasks.

Meetings with the steering group were held every other Thursday. Here, the project group gave an overview of the latest progress and discussed current issues. The project group had internal meetings at the end of every week, where a progress report was written. When the mandatory subject Systems Engineering was active, internal meetings and progress reports were every other week, due to less progress during those weeks.

A project plan was defined in the pre-project report. This was represented in a Gantt chart with individual tasks, planned start dates, and deadlines. The Gantt chart was continuously updated with progress and delays. The work hours were recorded and classified by task and group member. If a member cannot finish a task before the deadline, the rest of the group must be informed so that they can make decisions together based on that. Communication was considered an important part of the project.

4.2 Setup and workflow

This section describes the setup and workflow used to develop and execute the project. The first part describes how the Forte is compiled to run either as a local version or compiled for a Wago PLC. Furthermore, this process is simplified and semi-automated with a custom program. The method used to identify the addressing of the I/O modules is then described before explaining how and why version control is used. The workflow used to make and implement mathematical models is defined in the next part and consists of using Simulink to discretise and export the model to Structured Text. In the last part, the method used to document and execute the wiring of the system is explained.

4.2.1 Building Forte

4.2.1.1 Building of FORTE for Wago PFC200

To use Forte on a Wago PFC200 controller, multiple steps must be followed before the FORTE runtime can run. The general setup is based on the guide made by 4diac in [28] and the guide [29] to set up the firmware SDK for the PLC. There were multiple deviations from these guides, and the complete process is summarised below.

Ubuntu 20.04 LTS was used for the setup with all packages listed in Table 3.6 installed. The SDK Firmware is made available by Wago on Github at [29] and is cloned into the directory *$HOME/wago/ptxdist*. However, to build and use this firmware, the correct cross-compilation toolchain and the ptxdist tool must be installed. Both these tools are made available by Wago and can be easily cloned and configured by running the commands shown in Listing 4.1.

Listing 4.1: Installing ptxdist and toolchain. Adapted from [29].

```
$ # Make the directory for the toolchain and clone the repo
$ sudo mkdir -p /opt/gcc-Toolchain-2019.12/
$ sudo git clone https://github.com/WAGO/gcc-toolchain-2019.12-precompiled.git
    /opt/gcc-Toolchain-2019.12/
$ # Clone and check the configuration of ptxdist
$ git clone http://github.com/wago/ptxdist.git $HOME/ptxdist
$ cd $HOME/ptxdist
$ ./configure        # Check for any missing packages
$ make
$ sudo make install  # Installs the environment and create sym-links
```

When ptxdist is correctly installed, some configuration must be done. This includes setting the software configuration, hardware platform and toolchain. When the configuration is done, the configuration can be tested and if successful the firmware can be built. These steps are summarised in Listing 4.4 and should be executed in the project directory (i.e. $HOME/wago/ptxproj).

Listing 4.2: Configuring, testing and compiling the firmware SDK. Adapted from [29].

```
$ # Configure the SW configuration, platform and toolchain
$ ptxdist select configs/wago-pfcXXX/ptxconfig_pfc_g2
$ ptxdist platform configs/wago-pfcXXX/platformconfig
$ ptxdist toolchain /opt/gcc-Toolchain-2019.12/arm-linux-gnueabihf/bin/
$ # test the configuration
$ ptxdist menu       # Exit by selecting [Exit]
$ # Build and compile the firmware, this may take several hours.
$ ptxdist go -q
```

This compiled version of the firmware does not contain the Forte runtime. The rule files provided in the forte source code (available at [29]) can be used with minor changes to add the forte runtime to the firmware. The rule files located in buildsupport/wago_pfc200/ in the source are copied to the ptx project folders rules directory (i.e. $HOME/wago/project/rules/). In the *forte_wago.make*, the following changes have to be made:

1. All double spaces (used to signify a indent in the code) has to be changed to tab, since ptxdist do not read double spaces as a valid indent.

2. Set the correct path (line 27) to the Forte source code. If the source code is in the home directory with the name forte_src the line would be:

   ```
   FORTE_WAGO_URL   := file://$HOME/forte_src
   ```

3. Set the correct CMake options (line 49) for the project (see Section 4.2.1.2 for more information). The following configuration was used in the project to enable custom function blocks and the IEC61131-3 library:

   ```
   FORTE_WAGO_CONF_OPT := $(CROSS_CMAKE_USR) -DFORTE_COM_ETH:BOOL=ON
       -DFORTE_COM_FBDK:BOOL=ON -DFORTE_COM_LOCAL:BOOL=ON
       -DFORTE_ARCHITECTURE:STRING=Posix -DFORTE_MODULE_CONVERT:BOOL=ON
       -DFORTE_TRACE_EVENTS:BOOL=OFF -DFORTE_MODULE_WagoKbus=ON
       -DFORTE_MODULE_IEC61131=ON
       -DFORTE_EXTERNAL_MODULES_DIRECTORY=$HOME/FORTE_dev/ext_modules
       -DFORTE_MODULE_EXTERNAL_mconLib=ON
   ```

Some changes need to be done in the generated code that was compiled in the first compilation. The files that need to be changed is located in the directory
wago/ptxproj/platform-wago-pfcXXX/sysroot-target/usr/include/ and named:

1. ldkc_kbus_information.h

2. ldkc_kbus_register_communication.h

Line 25 must be edited for both files to:

```
#include <OsLinux/OsCommon.h>
```

After these changes, the Forte runtime can be compiled to an executable with the last command listed in Listing 4.3. The compiled executable can then be found in *$HOME/wago/ptxproj/platform-wago-pfcXXX/build-target/forte_wago-1.6.2-build/src* with the name *forte*. The compiled firmware SDK can now be transferred to the PLC. This can for example be done by exporting the firmware to an SD-card and booting the PLC from the SD-card by following the guide supplied by Wago, which is described in Section 5 and onward in [29].

The Forte executable must be rebuilt with any new or changed function blocks, this can be done with the commands in Listing 4.3. Thus, the executable must also be exchanged on the PLC to test the new version.

Listing 4.3: Commands to rebuild the Forte runtime for wago. Adapted from [29].

```
$ # Execute the commands in the ptx project directory
$ ptxdist clean forte_wago
$ ptxdist targetinstall forte_wago
```

4.2.1.2 Building of FORTE for local simulations

There are two main ways of running a local version of the Forte runtime. The first way is to use one of the pre-compiled versions available on 4diacs download page at [30]. These are versions for both Windows and Unix environments. The main disadvantage of using these is that they are not compatible with custom- and self-made function blocks. Therefore, the functionality can be limited depending on use cases.

The second way is to build the runtime directly using the sourcecode. This provides greater freedom since it is possible to use custom parameters and external libraries. CMake is used to setup the parameters for the compiler. It is possible to use both the GUI version and the CLI version. These have both advantages and disadvantages. The GUI version makes it is easy to see and understand all the different parameters. For the CLI version, the main advantage is that it is not needed to interact with a GUI, but everything can be defined with a single command. Both methods have been used extensively for different use cases in this project.

The CMake-GUI is an easy and visual way to configure a CMake environment with custom settings and generate the build files. In Figure 4.1, an example setup is shown, where both a

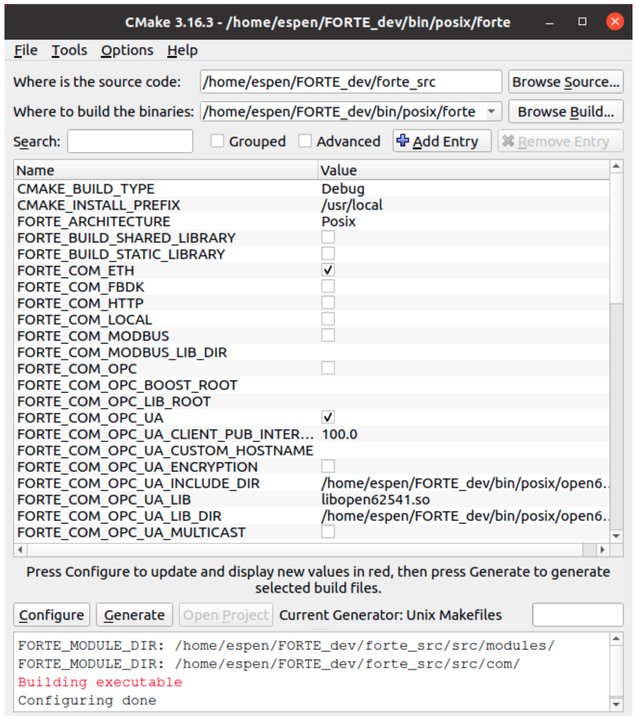


Figure 4.1: Example of using the CMake-GUI.

source path and a build path are specified together with a the CMake variables that describe the build options.

All the parameters shown when using the GUI can be set when using the CLI command of CMake. The same configuration that is in Figure 4.1 can be done with a single command as shown in Listing 4.4 if all the parameters and variables are known. From the CMake-GUI it is possible to extract the parameters that is needed for the configuration. For any parameter shown in the GUI the corresponding parameter in the CLI have the same name but uses the *-D* prefix and the datatype is defined if applicable after the name. For example the *FORTE_COM_ETH* that is shown in Figure 4.1 is transferred into *-DFORTE_COM_ETH:BOOL=ON* in Listing 4.4.

Listing 4.4: CMake command line example.

```
$ cmake -DCMAKE_BUILD_TYPE:STRING=Debug \
    -DFORTE_COM_ETH:BOOL=ON -DFORTE_COM_FBDK:BOOL=ON \
    -DFORTE_COM_LOCAL:BOOL=ON -DFORTE_COM_RAW:BOOL=ON \
    -DFORTE_ARCHITECTURE:STRING=Posix -DFORTE_MODULE_CONVERT:BOOL=ON \
    -DFORTE_TRACE_EVENTS:BOOL=OFF -DFORTE_MODULE_IEC61131=ON \
```

```
-DFORTE_EXTERNAL_MODULES_DIRECTORY=${HOME}/FORTE_dev/ext_modules \
-DFORTE_MODULE_EXTERNAL_mconLib=ON -DFORTE_COM_OPC_UA=ON \
-DFORTE_COM_OPC_UA_INCLUDE_DIR=${HOME}/FORTE_dev/open62541/build \
-DFORTE_COM_OPC_UA_LIB=libopen62541.so \
-DFORTE_COM_OPC_UA_LIB_DIR=${HOME}/FORTE_dev/open62541/build/bin \
-DFORTE_COM_OPC_UA_PORT=4840 -DFORTE_COM_OPC_UA_CLIENT_PUB_INTERVAL=100.0 \
-S ${HOME}/FORTE_dev/forte_src/ -B ${HOME}/FORTE_dev/bin/posix/
```

4.2.2 Program to simplify

When actively developing and testing custom function blocks, it is time-consuming to go through the complete building process for every iteration. Additionally, the process is more complex based on the different procedures for each case, like whether to test externally on a Wago PLC or locally. A custom bash-script was developed to simplify and streamline the process of building the different versions. This script is especially designed for this project but should also apply to some degree for other projects. The script has multiple modes and options for different functionality.

- The first mode is to build a local version of Forte using the standard CMake options for the project with custom function blocks. This mode use the CMake CLI command with a predefined source and destination path.

- The second mode is to send a new version of the Forte runtime to a connected PLC. Ptxdist is used to build this version of the runtime using the options defined in the rule file described in section 4.2.1.1. After the new version is built, the user is asked if the newly built program should be sent to the PLC controller. If selected, the user is asked to input the ssh-hostname to the controller, and if left empty, a default ssh-hostname is used. The built program is then sent to the PLC using the ssh-protocol using the scp command.

- The third mode is to send the bootfiles for the Forte instance. The bootfiles is used to autostart a Forte instance with a prebuilt program, so it is not necessary to deploy a program to a Forte instance after startup.

4.2.3 Addressing I/O modules

To use I/O modules in the project, it is important to understand how modules are addressed and accessed in the 4diac IDE. There were limited documentation for how the modules are addressed for the Wago PLC when using Forte, but there are a few examples of how I/O is addressed on other platforms. This information can be used as a starting point, but there is no guarantee that it is transferable, but the syntax should be the same. Multiple tests were executed that tests different types of I/O modules to check if there are any special cases that need to be accounted for and if they work as expected.

It is important that all the modules that are used in the project are tested and verified to make sure they work. Further, between tests, the order of the modules was changed to help identify patterns in the addressing. For any problems that occurred with functionality or were addressed in any modules, more testing was done to identify the root cause.

4.2.4 Version control

It was decided to use version control in this project because it is a great tool for working on a large project. It helps keep track of all changes in the project and even makes it possible to revert to a previous version if wanted. The version control platform used in this project is Github, a hosting service for the version control system Git. Using Github makes it easier to share the project between the group members without needing to send files back and forth.

All project files in Simulink, 4diac, Python, and more, were hosted on Github. On Github was a main branch. A new branch was started from the main branch for every feature addition or fix, and the work would be done on that branch. The branches were named after the type of work it was meant for and then more specifically, what it was for. For example, if a branch was created to make a function block for azimuth, it could be called *feature/FB-azimuth*. When the work on the branch was completed and tested, it was merged into the main branch with the review of at least one other group member. This way the main branch will always contain a stable and tested version of the project.

4.2.5 Simulink, discretisation and PLC code export

The dynamic models for the propeller, pitch and azimuth were created in Simulink. This will be explained in more detail in Section 4.3. After the models were created, they were exported from Simulink to PLC code, a software that exports Simulink files as structured text or ladder diagrams. After the files were exported as ST they could be imported into a function block in 4diac of the Simple type. For the models to be exportable to PLC code they needed to be discrete. It was found that the best and easiest way to get a good Simulink model that was exportable was to first make it continuous and then discretise it using the *Simulink Model Discretizer*.

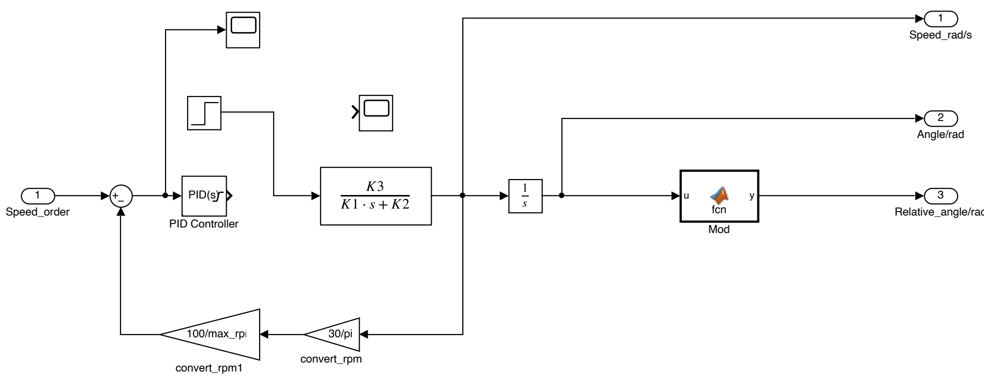


Figure 4.2: The continuous control system for azimuth in Simulink.

After making a continuous model in Simulink, such as the azimuth model in Figure 4.2, the *Simulink Model Discretizer* app seen in Figure 4.3 was opened.

First, what was going to be discretised was chosen on the left side. It is possible to discretise one block at a time or all continuous blocks in the model simultaneously. Next the settings on the right side were chosen. The Tustin method was used. This was mainly because compared to the other available discretisation methods, the results of this method were proved by trial and error to give the most similar responses in the models before and after discretisation. All the integrators in the model were set to use the forward Euler method for PLC Code compatibility. The propeller and azimuth models were discretised with 0.1 sample time, but the pitch model was set to 0.01. The pitch model needed a smaller sample time because the model response became very unstable with 0.1 sample time. In the *Replace current selection with:* section, the *New discrete subsystem* parameter was chosen.

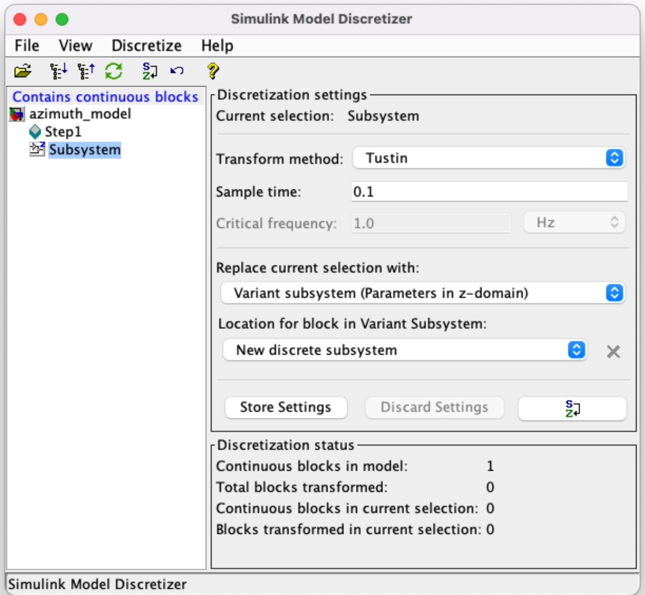


Figure 4.3: The *Simulink Model Discretizer* app.

After running the *Simulink Model Discretizer* the continuous blocks were converted to discrete. Figure 4.4 shows a discretised system.

4.2.6 Wiring

When doing any wiring, it is essential to document the connections. The primary documentation used in this project will be the signal lists. From the documentation supplied by Kongsberg, signal lists for Mcon were included. The signal lists and documentation were used throughout the project to better understand the system and how to develop the thruster simulator to be compatible with it. A signal list is developed or updated to document the work when planning and performing any wiring. In the signal list, both the source and destination are listed with a description of the signal function. The signal list will also give an overview of the address through which the signal can be accessed in 4diac. This will help streamline the programming of the system.

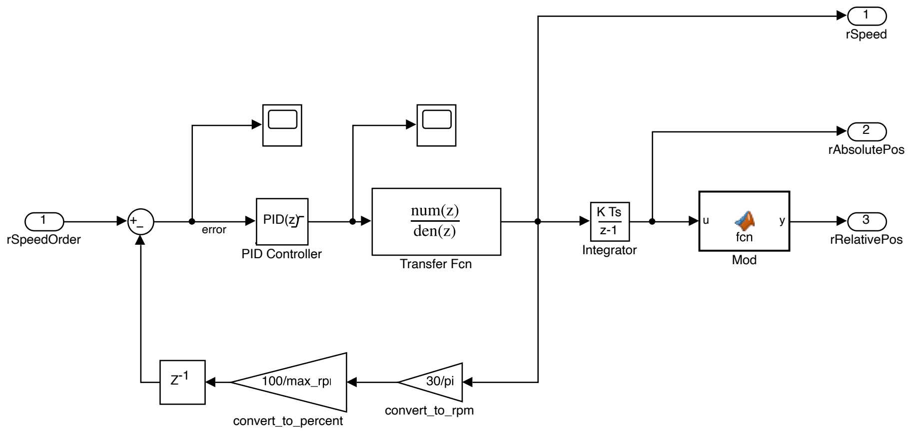


Figure 4.4: The continuous control system for azimuth discretised using the *Simulink Model Discretizer*.

4.3 Modelling and programming

4.3.1 Porting of the old program

The old PLC program in Automation Studio was examined in relation to the signal list for the Mcon system. An overview of the main functionalities and logic could be made based on this examination. Then a plan was sketched for implementing the program into 4diac. Since 4diac is based on the IEC 61499 standard while Automation Studio is based on the IEC 61131-3 standard, the program could not simply be ported over as is. The sketched plan needed to include restructuring, i.e. how the program should be divided into function blocks. A primary plan was made, but it was discovered to not work so well in execution soon after starting it. This was due to a misinterpretation of the purpose of some parts of the program, mainly the functionality that is not a part of the actual thruster simulator. Therefore, a secondary plan was made that prioritised what each part of the program should simulate. his plan was then executed.

The secondary plan described the new function blocks that needed to be created. The function blocks were divided by what equipment they should simulate. All signals necessary to simulate a pump belong to one function block, all signals necessary to simulate a thruster belong to one function block, and so on. The dynamic model of the thruster in Matlab and Simulink from last year's project was discretized and exported as a PLC project. This file was then imported into a function block in 4diac. This way, the PLC program avoids communicating with another program. Version control was used during the programming to have a complete overview of the development and safer development. For this, Github was used. When the secondary plan was executed, the first version of the main program for the thruster simulator was created.

The program was then uploaded to the Wago PLC, connected to the Mcon system. The test was performed from the Mcon system by executing the start/stop of the pump and thruster and checking that other operations performed as expected. The bugs that were discovered during this testing were noted and fixed.

4.3.2 New general thruster model

It was planned to make a new dynamic thruster model in Simulink that should include a controllable pitch. This model should calculate pitch feedback and take the pitch into the calculation of the rpm and thrust. At first, it was supposed to be based on a model from last year's project, but it became evident that some improvement was necessary during the recreation. After some research, it was decided to scrap this model and instead create a new one based on the model described in chapter 2.2.3 by Healey et al. [10]. This model gives the rpm and pitch response compared to order and the influence of pitch and water dynamics on the rpm. Because of this, the accuracy of the FPP and CPP dynamic models was considered improved if this model was used, as was originally a planned task for later in the project.

In Simulink, the program was divided into two subsystems, to be imported into two separate function blocks in 4diac. The first one was for pitch dynamics and was made with input for pitch order in percentage and output for pitch feedback in radians. The subsystem was made with a closed-loop control system with a PID controller, transfer function, and feedback. The transfer function was not made from the actual pitch system. A first-order transfer function was adjusted by trial and error to make the response similar to the expected response of a real pitch system. This control system is mainly based on the dynamic pitch model from last year's project but with some adaptations.

The second subsystem was for propeller dynamics (excluding pitch dynamics) and had input for rpm order in percentage and pitch feedback in radians. The subsystem was made with rpm feedback in percentage and thrust in newton as outputs. Two closed-loop control systems and a Matlab function for calculations were made inside the subsystem. The first control system is for the rpm. While the pitch control system was made based on an estimate of the transfer function for the system, it was chosen to base the rpm control system on the differential equation 2.26 for the rotational speed of the DC motor ω_m. Using the gear ratio N, this was converted to the rotational speed of the propeller ω_p. The rpm control system was made with a PID controller and several feedback connections.

The other control system is for the water dynamics. This was modelled after the differential equation 2.26 for fluid velocity U_a. It was made with a feedback connection but no PID regulator.

A Matlab function was made to execute the calculations for thrust and torque from equations 2.27 - 2.34. The function takes inputs from the propeller dynamics and water dynamics, as well as the pitch dynamics. The thrust and torque on the propeller were set as outputs.

The selection of parameters was made based on the values used in [10] for the water dynamics and extracted from the factsheet for the tunnel thruster (PM 2000). All the other parameters were selected to get the desired response from the system.

4.3.3 Azimuth implementation

The azimuth function makes it possible to rotate the thruster about the z-axis. This makes it possible to direct the thrust from the thruster in any horizontal direction, assuming it can have a full 360°range on the azimuth function. From the documentation supplied by Kongsberg the following features was identified to control the azimuth function and must therefore be simulated to be checked in a FAT-test:

- The Mcon system controls the azimuth rotation by giving commands using two PWM signals, one for counterclockwise and one for clockwise rotation. This PWM signal is supplied by a B&R PWM module.
- There is an encoder to read the rotation angle of the thruster. The Mcon system gathers the feedback from the encoder by CAN-bus using the CANopen protocol.

4.3.3.1 Reading PWM-signal

There are multiple ways to implement the reading of a PWM signal, and any solution should be considered based on the characteristics of the PWM signal. The data sheet for the PWM module used in the documentation (X20MM2436)[31] identified that the PWM signal has a voltage between 24V and 39V and can have a frequency between 1Hz and 50kHz based on the settings used. To simplify, it is assumed that the PWM signal have a voltage of 24V and a frequency of 2kHz.

It was decided to use a 0-10V analogue input module to read the PWM signal. Thus, the signal must be transformed from a PWM signal to a voltage between 0V and 10V based on the duty cycle of the signal. To achieve this, a simple electronic circuit was designed using an RC-filter and a voltage divider as described in Chapter 2.2.1. From the transfer function of this circuit between the input voltage V_s and the output voltage V_1, rewritten in 5.2, the frequency response of the system can be decided based on the values selected for R_1, R_2 and C.

$$\frac{V_1(s)}{V_s(s)} = \frac{\frac{R_2}{R_1+R_2}}{\frac{R_1 R_2}{R_1+R_2} C s + 1} \tag{4.1}$$

The steady state gain for the circuit is decided by the upper fraction in the transfer function:

$$\text{gain} = \frac{R_2}{R_1 + R_2} \tag{4.2}$$

Since the circuit must transform 24V to 10V and the output voltage is decided by the input voltage times the gain. The gain can be calculated to be:

$$\text{gain} = \frac{10V}{24V} = \frac{5}{12} \tag{4.3}$$

and by substituting this into 4.2 to get 4.4 the relationship between R_1 and R_2 must be 4.5.

$$\frac{5}{12} = \frac{R_2}{R_1 + R_2} \tag{4.4}$$

$$R_1 = 1.4 R_2 \tag{4.5}$$

The frequency response of the circuit must sufficiently dampen high frequencies so that the output voltage is stable with a high-frequency input voltage. To decide the values of the components, an arbitrary value was selected for R_2 and R_1 was then 1.4 times larger. The capacitor was then selected based on a simulation of the transfer function with a 2kHz input signal.

When the components are decided, the complete circuit is simulated in Multisim to check the response before the circuit is tested and measured with a function generator and an oscilloscope. The function generator that was available does not support an output of 24V. Hence, a new circuit was designed to amplify the signal to simulate the PWM signal. Two op-amps where used, one was used with positive feedback to amplify the signal. The second op-amp was used as a voltage follower to isolate the amplified signal from the feedback circuit to the first op-amp.

4.3.3.2 Dynamic model

The dynamic model used to simulate the feedback for the rotation is a simplified model that assumes a DC motor is used to directly drive the gear. It is further assumed that the order signal is the setpoint for the rotational velocity of the thruster. Hence, the rotational velocity is the control variable of the model even though the feedback is the relative rotation of the thruster. From section 2.2.2 the following transfer function for a DC-motor was derived:

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{\frac{K_t}{R_a J_m}}{s + \frac{D_m}{J_m} + \frac{K_t K_b}{R_a J_m}} \tag{4.6}$$

The parameters in the transfer function was selected based on crude estimation, and together with the PID-parameters was to get the desired response. The inertia was estimated by simplifying the thruster shape to be one horizontal cylinder placed on top of a vertical cylinder and can then be calculated with:

$$J_t = \frac{M_{D1} R_1^2}{4} + \frac{M_{D1} L_1^2}{12} + \frac{M_{D2} R_2^2}{2} \tag{4.7}$$

Where M_{D1} and M_{D2} is the mass of the horizontal cylinder and the vertical cylinder respectively. R_1 and R_2 denotes the radius of the cylinders and L_1 is the length of the horizontal cylinder. The total inertia of the motor inertia J_m would then be:

$$J_m = J_t + J_i \tag{4.8}$$

where J_i is the internal inertia of the motor. From the fact sheet for the Azipull thruster (AZP 85 PM L) some parameters was collected and estimated based on the drawings and the characteristics. The feedback that is used to calculate the controller error has to be scaled to percentage of the max speed since this is what the speed order.

4.3.3.3 Sending the feedback

Since the feedback should be coming from an encoder it is assumed that it is the relative rotation of the thruster. In other words the feedback should be given as the rotation of the thruster in for example the range 0 to 2π. Therefore the output from the system has to be integrated from the rotational velocity to the rotation position and be constraint to only show values between 0 to 2π.

4diac do not native support CAN-bus at the time of writing [32]. Thus, the CANopen protocol can not be directly implemented in the main program and an alternative solution has to be used. The proposed solution is to use an intermediary unit to implement the CANopen protocol. From the Wago PLC running the main program the feedback is given using a simple analogue output module. This analogue signal can then be read by a microcontroller which implements CANopen. Due to the complexity of implementing the CANopen protocol and the time limitation the feedback will be limited to only the analogue signal from the PLC.

4.3.4 Azipull implementation

A new system was made in 4diac to implement an azipull thruster. The system was very similar to the system for the tunnel thruster, but it uses the new thruster model for controllable pitch and the azimuth model. The pitch and azimuth orders should also be controllable from DP or Joystick mode. This means that it was necessary to make copies of most of the main function blocks and make them compatible with CPP and azimuth.

4.3.5 Swing-up implementation

Another system was made to implement the swing-up thruster. The swing-up also has controllable pitch and rotatable thrust direction like the azipull thruster, but in addition it has a swing-up functionality. Therefore the main app could be made from a copy of the azipull system, and only add the swing-up function block. To study the logic of the swing-up the signal list from a previous delivery from Kongsberg Maritime was used. It was important to understand the order of the sequence for the swing-up signals, but due to a delay in software and hardware updates the logic for the swing-up was not available and some assumptions had to be made.

A new function block for the swing-up functionality was made in 4diac. It modelled the response assumed expected to the Mcon system based on the orders. Using timers, the physical delay of locking/unlocking locking bolts and swinging the thruster up/down was simulated.

Mcon sends a PWM signal for lower order and lift order. This was simplified to a digital signal for several reasons. The PLC is not compatible with PWM signals, so it would need to be converted to 0-10 VDC with an RC circuit. But it was decided to test this only with the azimuth function. Thus, the logic was simplified to incorporate only digital lift and lower orders.

4.4 Complete system overview

The first plan for the overview of the system when including azimuth is shown in Figure 4.5. The Mcon system consists of levers, panel PCs with GUI, the main controllers and external I/O. These parts communicate using CANbus and Ethernet. The thruster simulator on the Wago was set up to run two instances of Forte simultaneously. This is described in greater detail in section 4.4.2. The instance of Forte with access to the K-Bus was connected to the Mcon external I/O by cable. This instance of Forte also communicates with the Docker instance of Forte, which sends necessary data to a computer through OPC UA for logging and FAT testing.

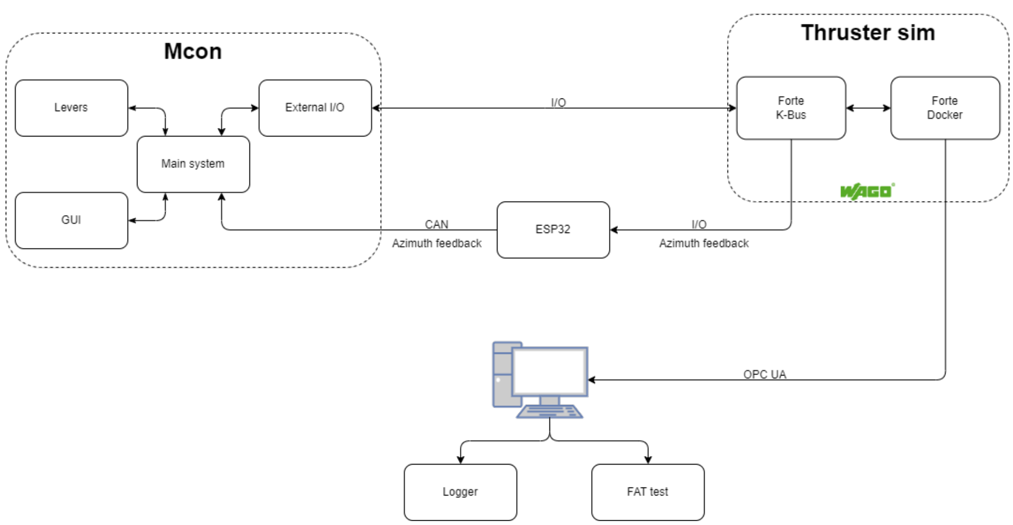


Figure 4.5: First iteration of the overall system and setup.

According to the initial plan, the azimuth feedback from the thruster simulator was going to be sent to an ESP32 via cabled I/O. Then the ESP32 would send the azimuth feedback over CANbus to the Mcon system. But as previously mentioned, this could not be done due to the complexity of the CANOpen protocol and the time limitation. At the same time there was a delay in the new Mcon software for the azipull and swing-up thrusters. Due to this, it was decided to instead test all the new signals including azimuth on the test jig, with the test jig replacing Mcon, as seen in Figure 4.6.

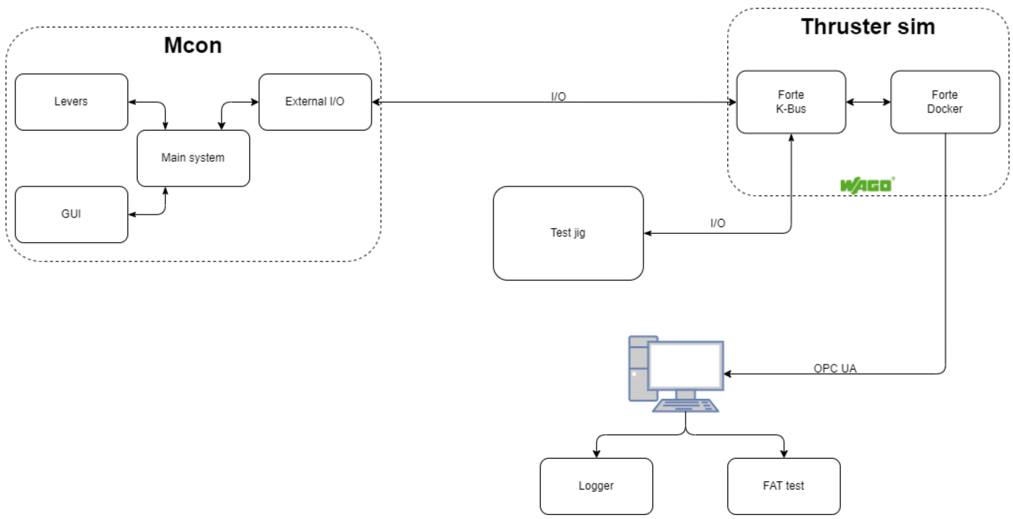


Figure 4.6: Final overview of the overall system and setup.

4.4.1 OPC communication

The OPC UA communication was mainly selected to achieve compatibility with the Java program used on the original platform. But it is also a communication protocol that have all the features that the project require which is the ability to read and set values over the protocol. This is also one of the protocols that is supported in 4diac using an external library called open62541. The Forte setup is explained in the next section.

From the 4diac program the OPC values is set up with a node name and ID. The node name was set based on what the value represent. Since the node object use a path and object structure it was defined so that when a node is defined it should reside within a *folder* which describe which part of the program the value belong to. Thus, the RPM feedback for the thruster simulator should have the name *rpmFeedback* and reside in the folder thrusterSim. The node ID that was chosen based on the function and what folder the node should be available in. There was used a 3 digit ID where the first was used for the folder, the second was used to describe the function and the third was used as a incremental value. The function digit was divided into multiple parts, as listed in Table 4.1. Notice that if the node was used as a readable point, the digit was under 5. If an OPC client was able to write the value it was set to 5 or above. This was done to make it easy to distinguish between different IDs.

Table 4.1: Description of the function digit used in the OPC ID (e.g. the Y in the ID: XYZ). The description is based on how an external client interacts with the ID.

Function digit	**Description**
1	Analogue value, readable
2	Digital value, readable
5	Analogue value, writable
6	Digital value, writable

4.4.2 Run two versions of Forte

There was considered multiple ways to enable OPC communication on the Wago PLC. The first way was to use ptxdist to cross-compile the open62541 library with the Forte instance. The second option was to run one instance on the PLC and one on a secondary computer, for instance a Raspberry Pi. It was also decided that it would be better not to add any extra material to the setup. Since the SOC in the Raspberry Pi 2B (ARM Cortex-A7) uses the same micro-architecture and instruction set as the Wago PFC200 (ARM Cortex A8) it was decided to test moving the setup to the PLC. This should be a viable solution, as the two SOCs are binary compatible. There would then be two native instances of Forte running on the PLC. The third solution was to use docker and use a containerised environment as described below. All three solutions should be tested, and the simplest solution should be used in the finished system.

From the source code of Forte available at [30], there is an example setup of a docker image. This was used and adapted to the needs of this system. The dockerfile is used to make the docker image and is adapted to run only the accompanying script for the initial setup of the image. There was also added a variable for extra run parameters when starting Forte. The accompanying script used by the dockerfile is adapted from the three script files, which are used in the source code and modified to only setup the necessary packages and libraries. This namely is the open62541 and mbedtls where both are required to implement OPC. Among the modifications, the compiling settings had to be changed to match and enable the arm architecture needed for the SOC used in the PLC.

The docker image was compiled on a Raspberry Pi to simplify the setup process. If the PLC had been used, it would have to be connected to the internet and is therefore undesirable. After the image was compiled, it could be exported and transferred to the PLC.

After the image is imported into the docker setup on the PLC, the docker container can start from the image. To start the container, the following command can be used if the image is named Forte with the version tag latest:

```
$ docker run forte:latest
```

This is the basic command, but it can also be extended to include more features and functionality by using flags and setting variables. Parameter —-*net=host* sets the container network configuration to use the native controller of the host machine. This allows the container to communicate over the network without allowing specific ports. This is helpful if the container communicates over random ports for example, the UDP broadcasting communication used in 4diac. It is possible to set the name of a container to avoid using a randomly generated name with the –*name* flag followed by the desired name. To share files between a container and the host, it is possible to mount folders with the -*v* flag. For setting the environment variables defined in the dockerfile the –env flag followed by the variable name equal some value. All these flags and parameters were used when starting the container and the command shown in Listing 4.5 was used in the startup script.

Listing 4.5: Command to start Forte container with parameters.

```
$ docker run -d \
    --net=host \
    --name forte_opc \
    -v /home/admin/forte-boot:/app/ \
    --env FORTE_EXTRA_OPTIONS="-c localhost:61530" \
    --env FORTE_BOOT_FILE=/app/thrusterSimulator_A200_OPC.fboot \
```

```
forte:latest
```

4.4.3 Gathering and showing results from thruster simulator

A python program was developed to collect and log results from the thruster simulator. The libraries used in the program are shown in table 3.5.

The main library in the program was the asyncua library, a combination of an OPC UA implementation in python and asyncio, a library to run asynchronous code, something which is not natively supported in python. This library only supports python versions no higher than 3.7, restricting the choice of python versions. Numpy, a library which includes methods and functions for advanced mathematics in python, was added to be able to store values in arrays and then further manipulate them as needed. Since only a maximum of five values was needed simultaneously, the program remained relatively simple.

The program would connect to the OPC UA server in the Forte docker container via IP, assign OPC nodes to a local variable, and then request the data from the node and write it to a variable. Then it would proceed by reading the system time in seconds and saving it to variable *startTime*. Arrays would then be created which would have the node values appended to them for each loop. Using the asynchronous functions from async the asyncio library allows all the value reading to be done in parallel, this allows the values to be collected at approximately the same time. The values would be appended for a number of times that was set for a while loop, and the loop would check the value of variable *counter* which would be incremented by one for each completed loop. Upon reaching the end of the while loop, it would subtract the time at the end of the loop to the *startTime* variable and save the difference to the *timer* array. When the while-loop counter reached its set value, it would exit the loop and write the array values to a csv file using the function *makeWriteableList*

The function *makeWriteableList*, takes four arguments and saves each argument in a single array, the initial argument were used for the array containing the time values. When all values are saved to the arrays, they would be combined into one large matrix, transposed and finally returned from the function. The matrix is transposed to have the values arranged vertically instead of horizontally.

A simple program in Python to make the graphs used in the results. First the log-file is imported and the correct data-ranges is selected before matplotlib is used to make the plots.

4.4.4 FAT testing in JAVA

The program used for semi-automatic FAT was made by Skarbø et al. in [2]. This program was made in Java and implements the OPC protocols to gather values from the thruster simulator. It was created to perform the FAT test on the finished system. To check whether the program was configured correctly, it was tested with the last system. Any problems that occur when using the program should be identified. If the root cause of a problem is to be found with the FAT-program, and not easily fixable, an alternative method can be used to test the OPC implementation on the thruster simulator.

Since the program would only be used to test the system it is not a priority to get the program to work, but it is a priority to have a theoretical compatibility with the program.

Chapter 5

Results

The following sections will list the results achieved in each part of the project. First results related to the overall project experience. Next an overview of the final physical setup will be given, as well as the final programs in the various software used. Next will be the thruster models and the related results from testing.

5.1 Project approach

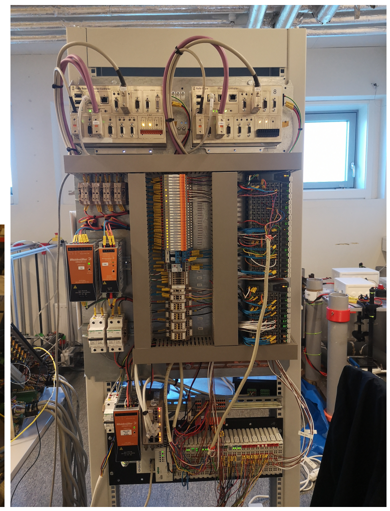
Before the project began, the group made a pre-project report planning the project and a Gantt diagram detailing the planned progress. These are available in Appendix A. The Gantt chart that was updated throughout the project and reflects the actual project flow is available in Appendix B together with the task overview and hour list. Progress reports describing the latest advances and problems were written regularly, and can be read in Appendix C. A meeting with the steering group was held every second week during the project period. The meeting minutes from these are available in Appendix D. During the project period the group also made a project pitch (Appendix E), a video presentation, and a poster (Appendix F).

The project was influenced by sickness in the group, including cases of Covid-19. The communication in the group was good, so the group was given notice when a member was sick.

5.2 Physical setup

A diagram of the full system overview can be seen in Figure 4.6. The Mcon test rig with the thruster simulator mounted at the bottom of the back can be seen in Figure 5.1b. The B&R PLC has been exchanged for a Wago PLC. Extra terminal blocks were mounted above the Wago PLC to supply 24 VDC. No changes has been made to the front of the Mcon test rig, as seen in Figure 5.1a.

(a) Mcon test rack from the front.



(b) Mcon test rack from the back.

The Wago PLC shown in Figure 5.2 was connected both to the Mcon system and to the test jig. The cables visible in the left of the image were a bit too short, but were not replaced due to the lab not ordering the requested cable. The Wago PLC as well as the I/O modules were given unit names. Details of the wiring and I/O addressing can be seen in the signal list in Appendix H.

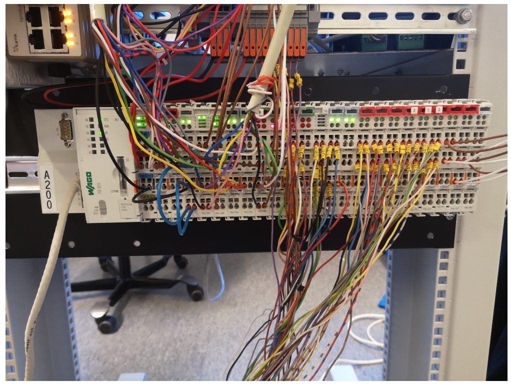


Figure 5.2: Wago PLC with I/O modules.

The test jig was used for the signals that were not available with the current Mcon software version. All signals were labeled on the jig. In Figure 5.3 the jig after the final testing is shown.

Figure 5.3: Test jig for simulating signals and feedback.

The test circuit for converting the PWM signal to an analogue signal was coupled on a bread board according to the schematic in Figure 5.13c. Figure 5.4 shows the full circuit, with the amplifier part of the circuit on the left and the RC-filter part on the right. The clamps on the right side of the circuit connect the analogue signal to the test jig, which connects it to the Wago PLC. The reason for sending the signal via the test jig was that the same signal was used both for CW and CCW rotation, and it was easier to switch back and forth on the test jig.

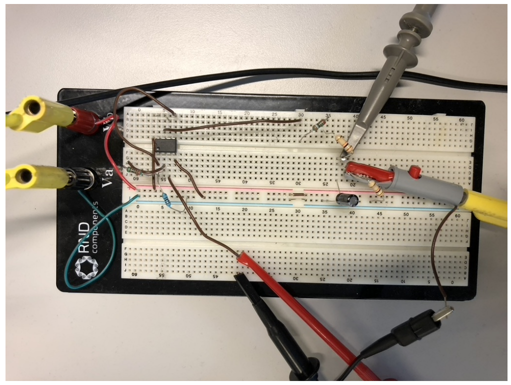


Figure 5.4: Test circuit for converting PWM signal to analogue.

5.3 Addressing of IO-modules

In total 14 rounds of systematic testing were completed to determine the I/O addressing system for Forte on Wago. Table 5.1 details what was tested in each test. Notice how the analogue inputs and outputs are always addressed with the same prefix 1 and 2 respectively. There are two exceptions where the analogue outputs were not working, see tests 7 and 10. It should be noted that the addressing of the digital inputs and outputs changes between some tests, for example, between tests 8 and 10. The last test also describes the finished PLC setup that are shown in Figure 5.2.

Table 5.1: Results of I/O addressing tests. N/A: Not applicable, NW: Not Working.

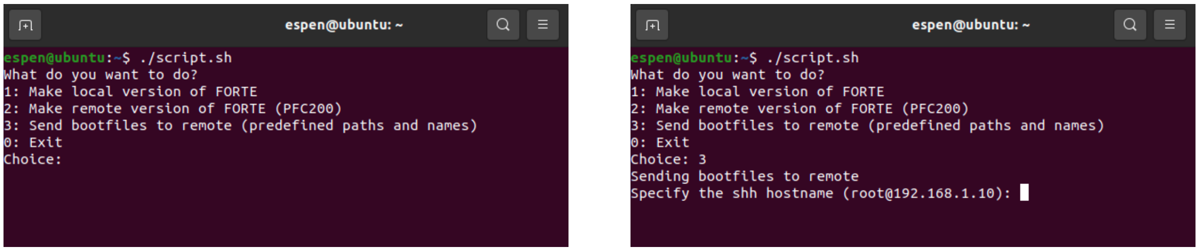
Test nr	Module 1	Module 2	Module 3	Module 4	Module 5	Module 6	Module 7	Module 8	Module 9	Module 10	Module 11	Module 12
1	750-1504 (DO) [0.0 - 0.15]	750-1405 (DI) [1.0 - 1.15]										
2	750-1504 (DO) [0.0 - 0.15]	750-1405 (DI) [1.0 - 1.15]	750-478 (AI) [1.0 - 1.1]	750-550 (AO) [2.0 - 2.1]								
3	750-1405 (DI) [0.0 - 0.15]	750-1504 (DO) [1.0 - 1.15]	750-478 (AI) [1.0 - 1.1]	750-557 (AO) [2.0 - 2.3]	750-530 (DO) [1.16 - 1.23]	750-430 (DI) [0.16 - 0.23]	750-478 (AI) [1.2 - 1.3]	750-550 (AO) [2.4 - 2.5]				
4	750-557 (AO) [2.0 - 2.3]	750-530 (DO) [1.0 - 1.7]	750-430 (DI) [2.0 - 2.7]	750-478 (AI) [1.0 - 1.1]	750-1504 (DO) [1.8 - 1.23]	750-478 (AI) [1.2 - 1.3]	750-1405 (DI) [2.8 - 2.23]	750-550 (AO) [2.4 - 2.5]				
5	750-557 (AO) [2.0 - 2.3]	750-530 (DO) [1.0 - 1.7]	750-478 (AI) [1.0 - 1.1]	750-1504 (DO) [1.8 - 1.23]	750-478 (AI) [1.2 - 1.3]	750-1405 (DI) [5.0 - 5.15]	750-430 (DI) [5.16 - 5.23]	750-550 (AO) [2.4 - 2.5]				
6	750-557 (AO) [2.0 - 2.3]	750-478 (AI) [1.0 - 1.1]	750-478 (AI) [1.2 - 1.3]	750-1405 (DI) [3.0 - 3.15]	750-430 (DI) [3.16 - 3.26]	750-530 (DO) [5.0 - 5.7]	750-1504 (DO) [5.7 - 5.23]	750-550 (AO) [2.4 - 2.5]				
7	750-513 (DO) [0.0 - 0.1]	750-513 (DO) [0.2 - 0.3]	750-513 (DO) [0.4 - 0.5]	750-626 (Filter) [N/A]	750-430 (DI) [3.0 - 3.7]	750-430 (DI) [3.8 - 3.15]	750-554 (AO) [NW]	750-554 (AO) [NW]	750-455 (AI) [1.0 - 1.3]	750-455 (AI) [1.4 - 1.7]		
8	750-430 (DI) [0.0 - 0.7]	750-530 (DO) [1.0 - 1.7]	750-554 (AO) [2.0 - 2.1]									
9	750-430 (DI) [0.0 - 0.7]	750-530 (DO) [1.0 - 1.7]	750-626 (Filter) [N/A]	750-554 (AO) [2.0 - 2.1]								
10	750-513 (DO) [0.0 - 0.1]	750-626 (Filter) [N/A]	750-430 (DI) [1.0 - 1.7]	750-530 (DO) [0.2 - 0.10]	750-554 (AO) [NW]							
11	750-626 (Filter) [N/A]	750-430 (DI) [0.0 - 0.7]	750-530 (DO) [1.0 - 1.7]	750-554 (AO) [2.0 - 2.1]	750-513 (DO) [1.8 - 1.9]							
12	750-517 (DO) [0.0 - 0.1]	750-517 (DO) [0.2 - 0.3]	750-430 (DI) [2.0 - 2.7]	750-530 (DO) [0.4 - 0.12]	750-626 (Filter) [N/A]	750-430 (DI) [2.8 - 2.15]	750-430 (DI) [2.16 - 2.23]	750-554 (AO) [2.0 - 2.1]	750-554 (AO) [2.2 - 2.3]	750-455 (AI) [1.0 - 1.3]	750-513 (DO) [0.13 - 0.14]	750-513 (DO) [0.15 - 0.16]
13	750-513 (DO) [0.0 - 0.1]	750-517 (DO) [0.2 - 0.3]	750-626 (Filter) [N/A]	750-430 (DI) [2.0 - 2.7]	750-430 (DI) [2.8 - 2.15]	750-554 (AO) [2.0 - 2.1]	750-554 (AO) [2.2 - 2.3]	750-455 (AI) [1.0 - 1.3]				
14	750-513 (DO) [0.0 - 0.1]	750-517 (DO) [0.2 - 0.3]	750-626 (Filter) [N/A]	750-430 (DI) [2.0 - 2.7]	750-430 (DI) [2.8 - 2.15]	750-554 (AO) [2.0 - 2.1]	750-554 (AO) [2.2 - 2.3]	750-455 (AI) [1.0 - 1.3]	750-455 (AI) [1.4 - 1.7]	750-459 (AI) [1.8 - 1.11]	750-459 (AI) [1.12 - 1.15]	750-430 (DI) [2.16 - 2.23]
14 cont.	**Module 13** 750-556 (AO) [2.4 - 2.7]	**Module 14** 750-556 (A0) [2.8 - 2.12]	**Module 15** 750-513 (DO) [0.4 - 0.5]	**Module 16** 750-513 (DO) [0.6 - 1.9]	**Module 17** 750-513 (DO) [0.7 - 0.8]	**Module 18** 750-513 (DO) [0.9 - 0.10]	**Module 19** 750-513 (DO) [0.10 - 0.11]	**Module 20** 750-513 (DO) [0.12 - 0.13]	**Module 21** 750-513 (DO) 0.14 - 0.15]	**Module 22** 750-517 (DO) [0.16 - 0.17]		

5.4 Programs and software setup

This section and its subsections describe the results achieved in each program used. First, the results for the script used to automate the workflow and the OPC logger are described. Then the PLC setup and the results for the docker setup are described. This also includes the results for testing and enabling OPC UA communication on the PLC. Next, the results from the 4diac setup and logic are described, this also includes a description of the results for selective function blocks. Finally the results from testing the semiautomatic FAT program are described.

5.4.1 Automated workflow script

The automated script used to simplify the workflow is available in Appendix Q and helped streamline the process. It was used as the main way to compile and distribute the Forte runtime to the PLC and for local simulation in the Linux environment. The script worked as intended and it was noted a faster development time after the script was taken into use. In Figure 5.5a the main menu is shown with the three implemented choices. For both choices where it is possible to send files to a remote (i.e. make remote version and send bootfiles to remote), the user is asked to enter a specified ssh hostname as shown in Figure 5.5b. When the user inputs a hostname the validity is checked with a regular expression. The address is only used if it is valid and if not the placeholder hostname *root@192.168.1.10* is used. It is not possible to change the paths that are used to send the files to the remote where */usr/bin/* is the path used for sending the Forte version and */home/admin/* is used for the bootfiles. When the bootfiles are sent the complete folder *$HOME/FORTE_dev/forte-boot/* is sent to the remote.



(a) (b)

Figure 5.5: The menu of the automated workflow script and the option to specify a shh hostname. (a) The menu with the three choices and exit. (b) The menu after selecting the third choice.

5.4.2 OPC logger

The python client is available in Appendix P.

To be able to receive values from the thruster simulator, a python client was used. The client would then connect with the docker container running Forte, and collect the published data from its assigned nodes through OPC UA. When the data was successfully received and written to a local variable, the program would write the variable values to a csv-file and then begin the next loop.

When collecting data from two nodes, the average execution time was 96.95ms with a standard deviation of 11ms. When collecting from five nodes, it would average 247.71 ms with a standard deviation of 12.57ms. By adding three nodes one can see that the execution time is increased by 2.55 times, making the data collection rate change from 10.3Hz to 4.03Hz.

A complimentary program was made.

5.4.3 PLC and Docker

All three solutions were tested to enable OPC UA communication from the PLC. From testing the first solution, it was noted that there was not found a way to enable the linking from the open62541 library, which can be enabled in the ptxdist menu, to the Forte instance. When testing the second solution, the Forte and open62541 were compiled on the Raspberry Pi and tested. It was noted that it worked as intended, and there were no inherent issues. After the setup was moved to the PLC, the program started, but could not access the external library open62541 and gave an error message. This was tested with the exact same file structure on the PLC and the Raspberry Pi.

For the third solution, the dockerfile and the accompanying bash-script used to compile the image is available in Appendix J. In the dockerfile, there are defined environmental variables to define the path to the bootfile and the name of the bootfile. The default behaviour of the variable is set to be as shown in Listing 5.1. There was also added a variable that can be used to add any extra options to the Forte runtime. These options are appended to the run command and are used in this project to set the port used for the communication. If the variable is not set in the start command to the docker container, the default behavior is used as shown in Listing 5.1.

Listing 5.1: Default definition of the environmental variables in the dockerfile.

```
ARG FORTE_BOOT_FILE=/usr/forte_boot/forte.fboot
ARG FORTE_EXTRA_OPTIONS=""
```

The docker image was compiled on a Raspberry Pi and transferred to the PLC. To auto-start, the docker container and the other version of Forte, the *auto_start* script in Appendix K was used. This started the docker version of Forte on port 61530 and the local version on port 61520. The script was linked to automatically start when the PLC is powered on. Both versions are set up to use the boot-files in the folder */home/admin/forte-boot* with the names: thrusterSimulator_A200_OPC.fboot and thrusterSimulator_A200.fboot. These names and paths can easily be changed in the *auto_start* script to correspond to other setups.

The startup time from the system received power to the thruster simulator was ready to be used was timed to be about 2 minutes.

5.4.4 4diac logic and setup

During the work in 4diac, a few issues were discovered. None of the issues resulted in a full stop for the progress but rather caused delays due to restarting the program or rethinking problem solutions. The reported issues are listed below.

- I/O addressing with Wago is neither intuitive nor documented
- No debugger
- Not compatible with functions or function blocks inside function blocks in Structured Text
- Autofill rarely works
- Print does not work on Linux desktop and algorithms cannot be printed at all
- Stability issues
  - Periodical crashes, sometimes very frequent (may be mostly on Windows desktop)
  - Some function blocks cause crashes if removed
  - Sometimes function blocks get "hung up" during runtime, i.e. getting an input but not writing the output as expected unless the function block is forced to run again

The full 4diac program is available in Appendix I. A system was created for each of the three thruster types; Tunnel thruster, Azipull thruster, and Swing-up Azimuthing thruster. Inside each system, there are two applications. One application for everything related to Joystick, DP, and visualisation, and one for the thruster simulation and control of the thruster.

The application for Joystick, DP and visualisation is named *dpJsVisuSim* and is almost identical for all three thruster systems. For the azimuthing CPP thrusters, some blocks were replaced with other versions of the same blocks that were updated to be compatible with adjustable pitch and azimuth degree.

The other application is named *thrusterSim*. This app contains the simulation of the thruster (see Section 5.4.4.1). It also contains the azimuth dynamics and logic for start/stop for the pump(s). In the Swing-up system it also contains the logic for the swing-up functionality, see Section 5.4.4.2.

Even though the IEC61499 standard is event-based, all main function blocks except for the one for visualisation were set to be triggered by a cyclic block. All the cycle times were set to 0.1 seconds.

5.4.4.1 Thruster simulator

The thruster simulation consists of several parts encased in a wrapper function block. Figure 5.6 shows a diagram of the execution order. The first part is for the start/stop of the thruster and setting the RPM order. If the thruster is off, the order is set to zero. The second part contains a dynamic thruster model imported from Simulink. This part calculates the RPM feedback based on the order.

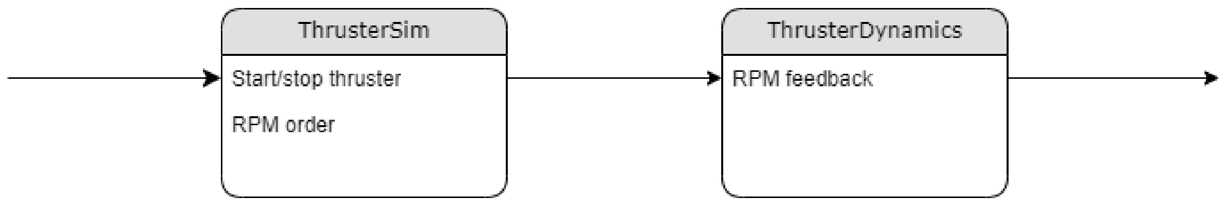


Figure 5.6: Diagram showing the execution order for the thruster simulator.

Another version of the wrapper function block was created when the controllable pitch was added to the thruster simulator. Figure 5.7 shows a diagram of the execution order for the CPP version of the wrapper. The ThrusterSim block is the same as in the first version. The PitchDynamics block contains a dynamic pitch model imported from Simulink and calculates the pitch feedback. In this version of the wrapper, the pitch influences the RPM feedback calculated in CPPDynamics. Since the sample time of the PitchDynamics was set to 0.01 seconds and the CPPDynamics have a sample time of 0.1 seconds, the PitchDynamics block is calculated ten times sequentially for every iteration of the wrapper. This makes the wrapper able to have a sample time of 0.1 seconds.

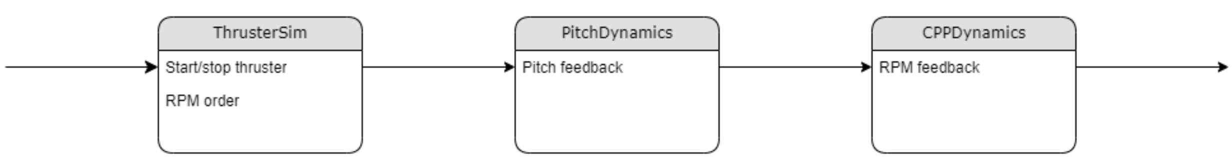


Figure 5.7: Diagram of the execution order for the thruster simulator with CPP.

5.4.4.2 Swing-up

A function block was made to handle the swing-up functionality. The logic was implemented using cases, which are shown in Figure 5.8 together with the case transitions.

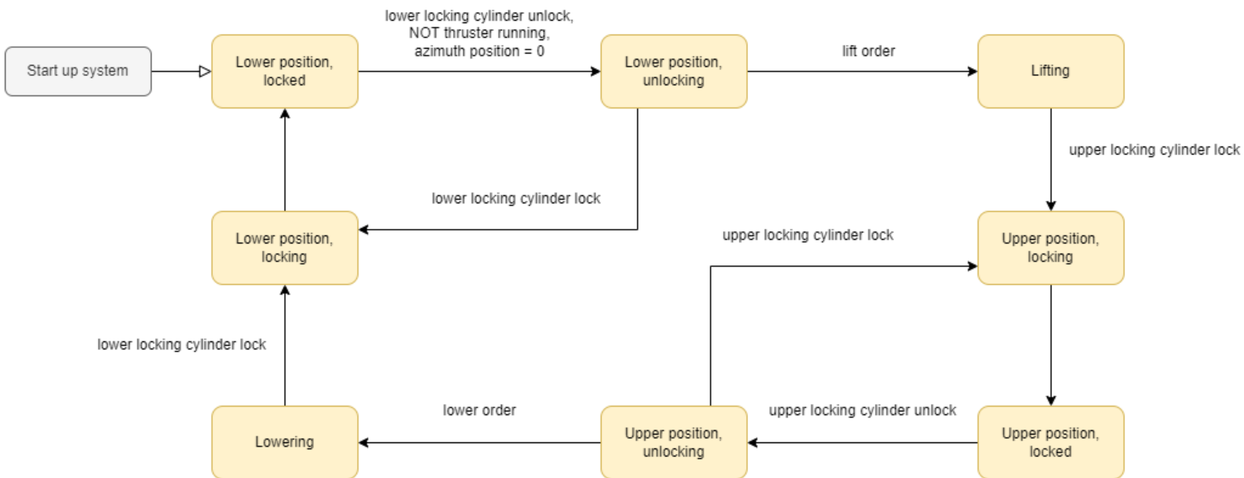


Figure 5.8: Logic of Swing-up function block.

5.4.5 FAT Java program

The semi-automatic program was not able to connect to the PLC program from last year's project. When using Wireshark to inspect the data-flow between the program and the controller, it was identified that the program did collect the available endpoints from the controller. With the IntelliJ debugger, it was further identified that the program did not select a valid endpoint which caused the program to not establish a valid OPC connection to the controller. Thus, it crashed when it tried to collect the OPC values. Fixing the problem was decided to be out of scope for the project.

5.5 Thruster models

This section contains all the results and measurements done of the thruster models. Results are included from the simulations of both the continuous and the discrete model in Simulink, and the logs collected from the implemented model running on the system. The azimuth model also includes the results from the conversion between the PWM signal to a 0-10V analogue signal. All the models that used PID-controllers had their parameters adjusted either by using the Simulink Tuner App or by intuition. In Table 5.2, the PID-parameters are summarised for the models.

Table 5.2: PID-parameter for the propeller models.

Model	P	I	D
Propeller	100	35	400
Pitch	0.0252	0.0047	0.0138
Azimuth	5.054	0.026	-1.140

5.5.1 Initial implementation of thruster

The initial implementation based on the thruster model from last year was tested with a step response on the discrete model. In Figure 5.9a, the response of the discrete model is shown. This model has a response time of 20 seconds to steady-state, mathing the continuous results shown in [2]. When the discrete model was implemented in 4diac, the response shown in Figure 5.9b was achieved and matches the discrete model. When changing the direction of the propeller, the response in Figure 5.9c and 5.9d was gathered for the discrete model in Simulink and the model implemented in the thruster simulator, respectively. Both responses have a settling time of 20 seconds without an overshoot. All the responses from the model implemented in the thruster simulator were generated by overriding the order signal from the Mcon system and collected with the OPC logger. The program used in Simulink is available in Appendix L with the accompanying Matlab script from [2].

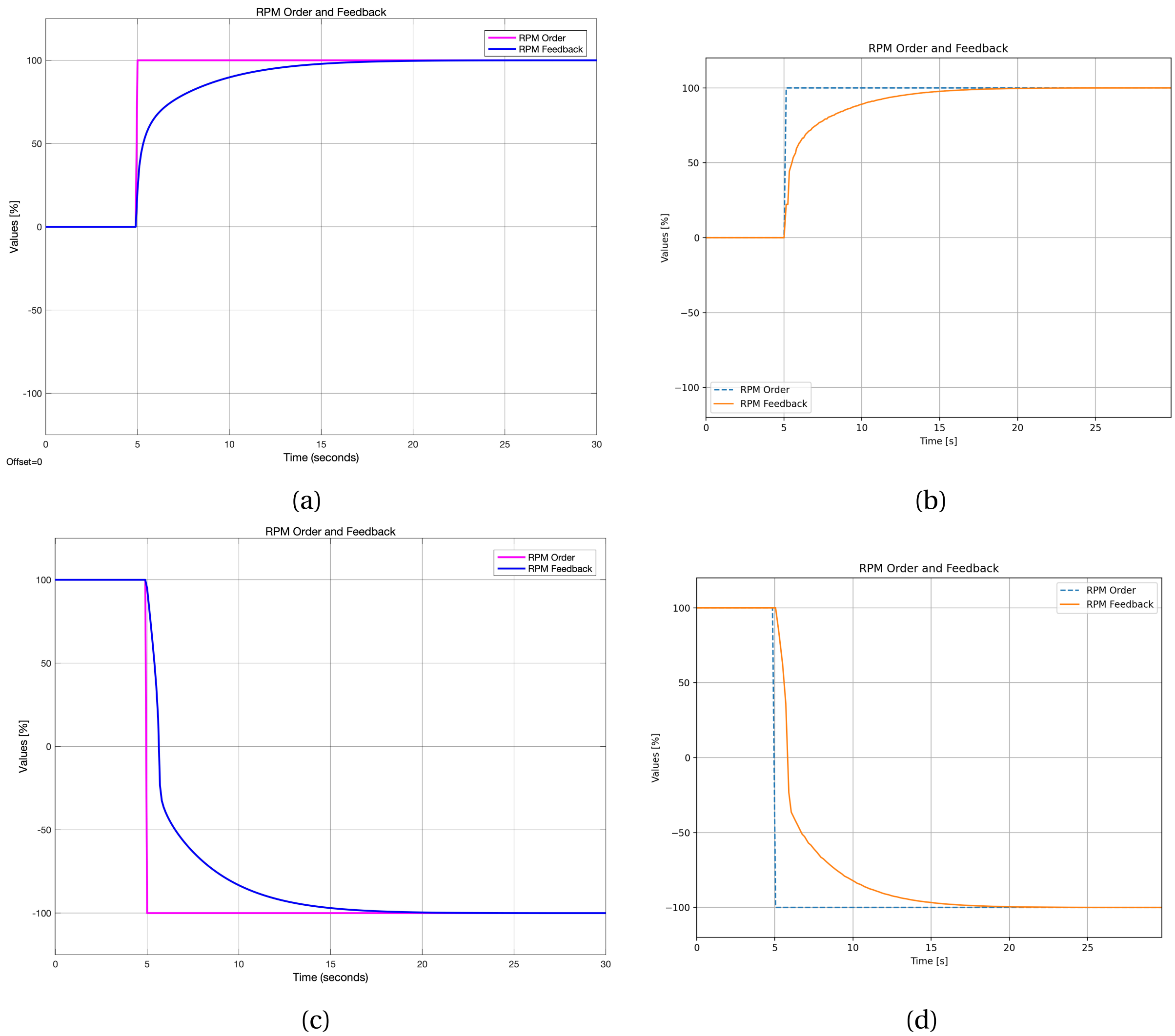
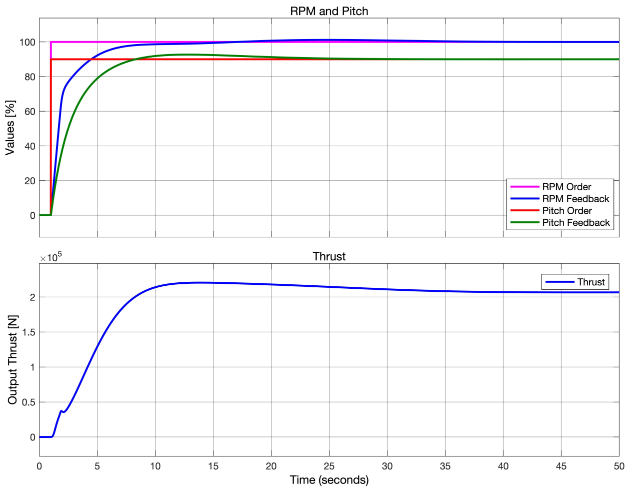
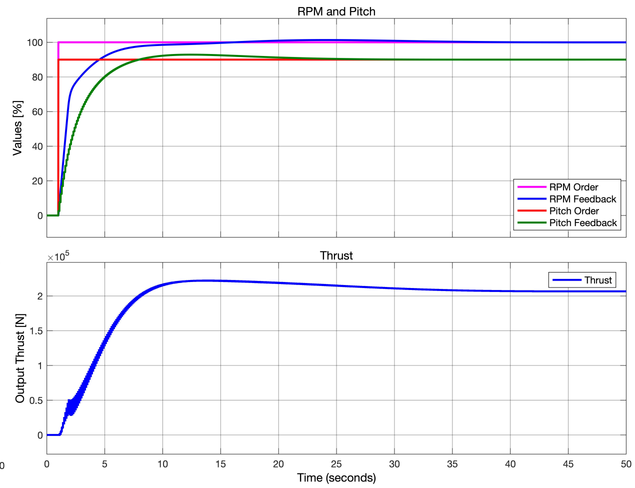


Figure 5.9: The response of the discrete model and the implemented system. (a) The discrete response for changing the order from 0% to 100%. (b) The discrete response for changing direction. (c) The system response for changing the order from 0% to 100% . (d) The system responds to changing directions.

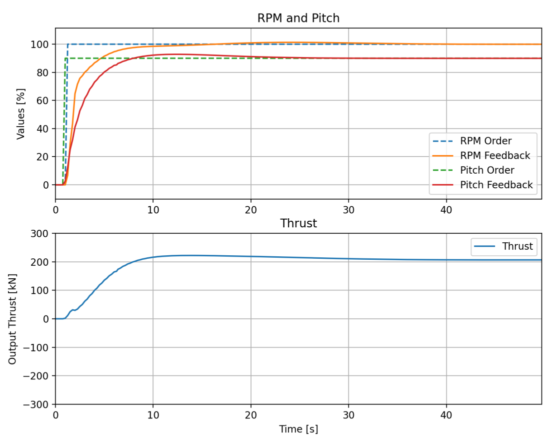
5.5.2 Propeller model



(a)



(b)



(c)

Figure 5.10: The step response when changing the RPM and pitch order to 100% and 90% respectively. (a) The continuous model in Simulink. (b) The discrete model in Simulink. (c) The response from the simulator.

The first subsystem that is used for the pitch dynamics was based of the first order transfer function:

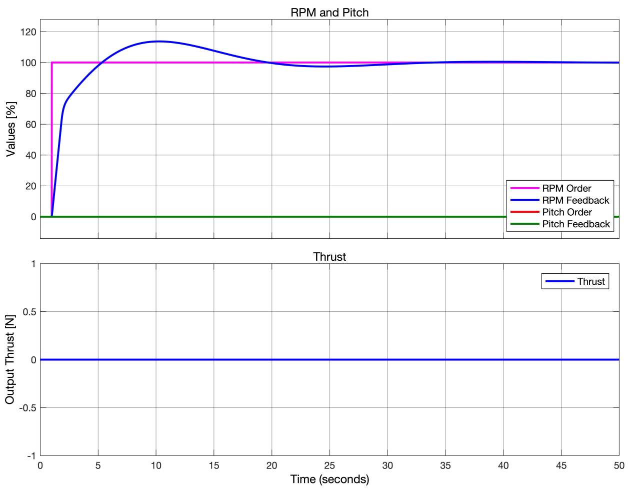
$$H_p(s) = \frac{1.5}{8s+1} \tag{5.1}$$

With a max pitch of $\pi/3$ selected for the feedback gain. For the second subsystem, the following constants and parameters were chosen from the factsheet for the tunnel thruster (PM 2000). The propeller diameter was 2 meters, with an estimated tunnel length of 2.5 meters. The inertia was estimated to be $4277.5 kgm^2$ but changed to $17110.0 kgm^2$ to get the desired response. The parameters for the effectively added mass ratio γ and the momentum coefficient $\Delta\beta$ was reused from [10] to be 0.5 and 0.2, respectively. In Appendix M, the Simulink models are available, and the accompanying Matlab script, which includes all the parameters.

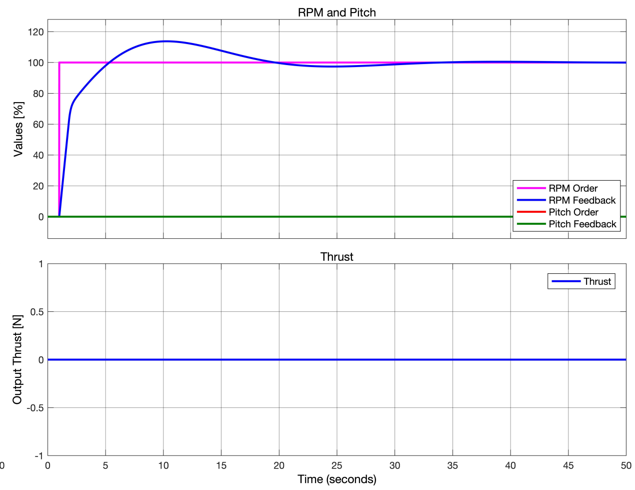
The propeller model was adjusted to achieve the desired response when the pitch was set to a step response of 90% and the RPM order was set to 100%. After adjustment, the PID-parameter in Table 5.2 was used. In Figure 5.10a, the response of the system after adjustment is shown. The continuous model was discretised with a sample time of 0.01s using the Tustin method.

Notice how the thrust fluctuates in the step response for the discrete system in Figure 5.10b. It was noted that with a higher sampling time than 0.01s, the thrust was completely unstable. When the model was implemented in 4diac, the response in Figure 5.10c was achieved by simultaneously forcing the RPM and pitch order to 100% and 90%, respectively. The response for RPM and pitch is similar for the continuous and discrete Simulink models and the implemented model in the thruster simulator. The RPM response have a fast response to reach 80% in two seconds before it slows down and reach 98% after nine seconds. In the response for the pitch the response is slower but there is first an overshoot of 92.8%, and when the pitch is lowered, there is an overshoot of 101.3% on the rpm response.

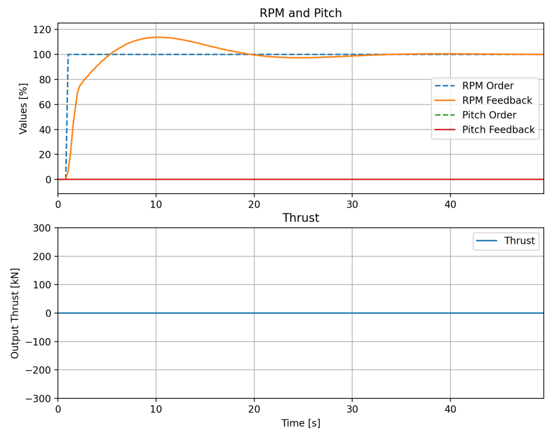
The model was also tested with a step signal applied to the model, and the pitch was set to zero. In Figure 5.11a, the response for the continuous model in Simulink is shown, and the discrete model in Figure 5.11b. For comparison, the response for the implemented thruster simulator is also included in Figure 5.11c. The three responses are similar and have the same fast response to 80% in 2 seconds as the above step response. However, there is an overshoot to 118% after 10 seconds. After 35 seconds (34 seconds after the step order), the response is stabilised at the order value.



(a)



(b)



(c)

Figure 5.11: The response when changing the RPM to 100% with a constant pitch of 0%. (a) The continuous model in Simulink. (b) The discrete model in Simulink. (c) The response from the simulator.

5.5.3 Azimuth

The results for the azimuth model are divided into two parts, with one for the signal conversion and one for the dynamic model and how the system responds to different orders.

5.5.3.1 PWM signal transformation

For the component in the RC-filter, it was selected to use one $14k\Omega$ and one $10k\Omega$ resistor for R_1 and R_2, respectively, and a capacitor of $2\mu F$ for C_1. When entering these values in the transfer

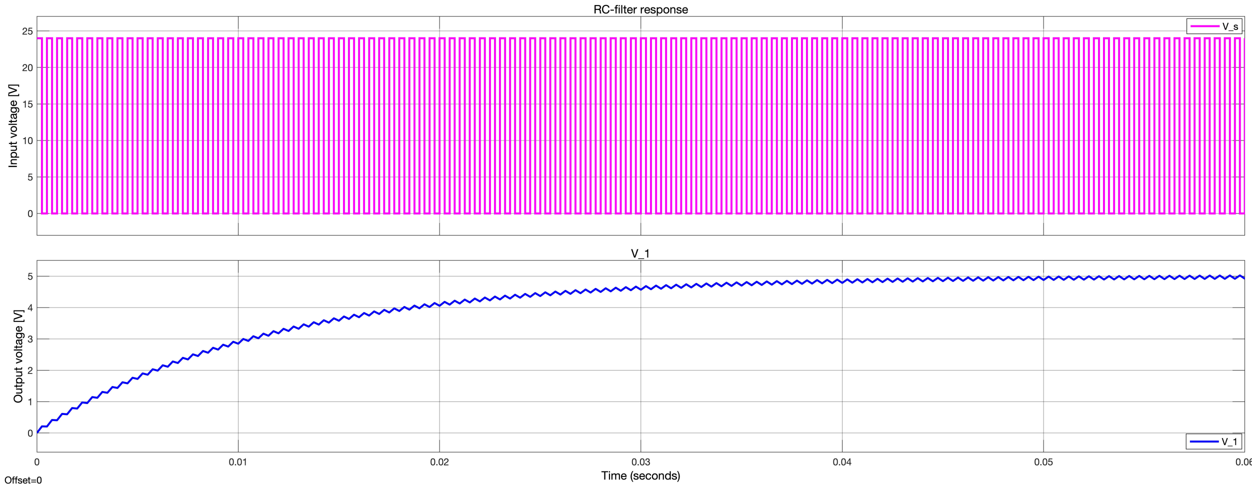


Figure 5.12: The simulated time response of the RC-filter. The top graph is the input signal to the transfer function, and the simulated response is in the bottom graph.

function in 5.2 the following transfer function was achieved:

$$\frac{V_1(s)}{V_s(s)} = \frac{0.4167}{0.01167s+1} \tag{5.2}$$

In Figure 5.12, the response of the transfer function is shown when the input signal was a $2kHz$ square-wave with an amplitude of 24. Notice how the output signal is a triangle wave with a peak to peak difference of about 0.1V.

The circuit was sketched and simulated in Multisim, as shown in Figure 5.13a and Figure 5.13b. There is a ripple 0.1V peak to peak in the simulated signal. Since the function generator could not output a 24V signal, a dual op-amp was used to boost the signal from 4V to 24V. This was simulated in Multisim, as shown in Figure 5.13c, and the response is as expected and the filtered signal has the same ripple of 0.1V, see Figure 5.13d.

The physical circuit was made from the schematic in Figure 5.13c, but there were chosen resistors from the E24 series. Thus, the R_1 was made of one $10k\Omega$ and one $3.9k\Omega$ resistor in series, and R_4 the $5k\Omega$ resistor, was switched to a $5.1k\Omega$ resistor. The signal at V_s when a square wave signal with a frequency of 2kHz and a duty cycle of 50% is shown in Figure 5.14a. Note how the signal gets a ramp up and down time when it is boosted by the op-amp.

The filtered signal V_1 with the same input signal of 2kHz is shown in Figure 5.14b.

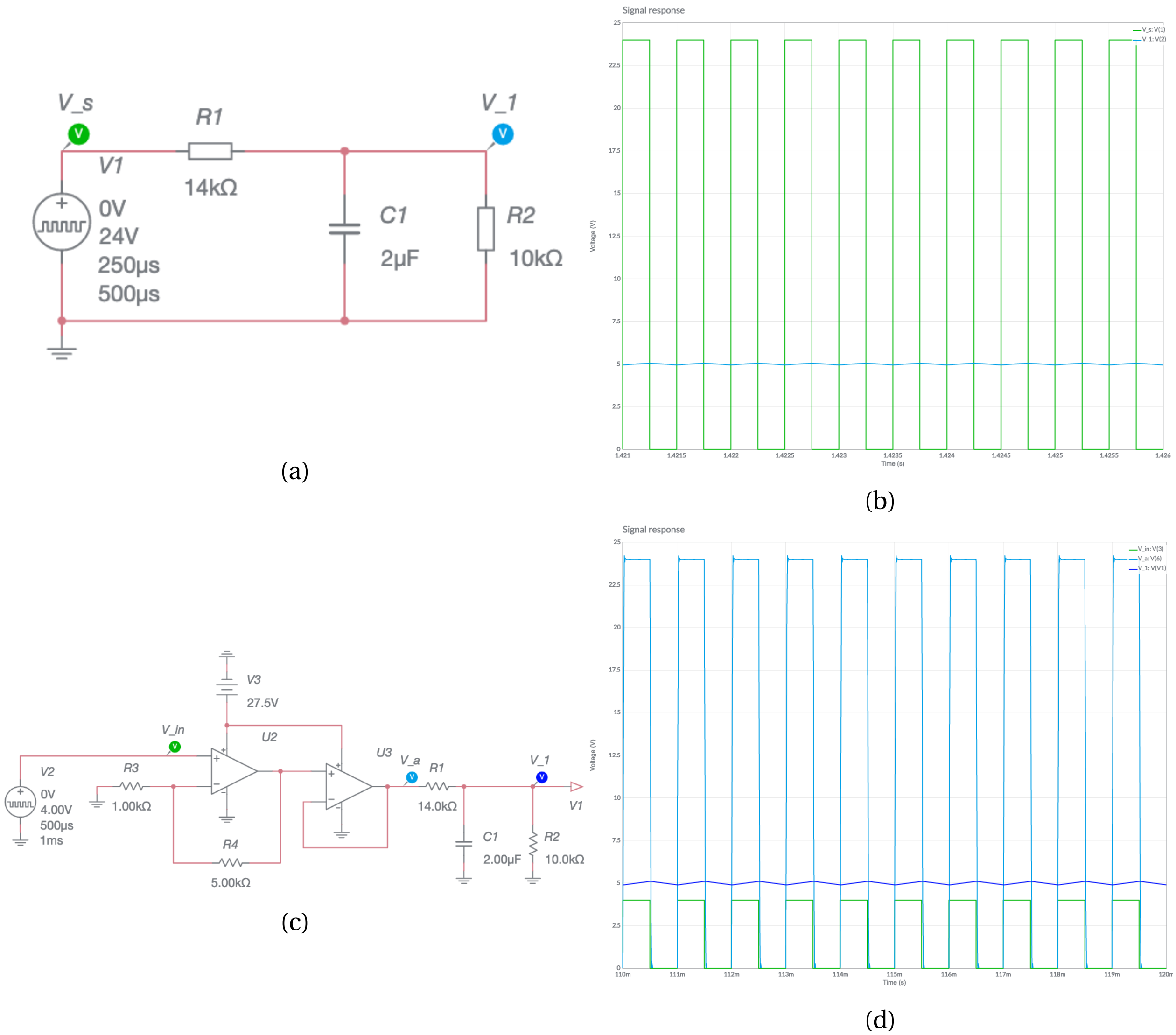
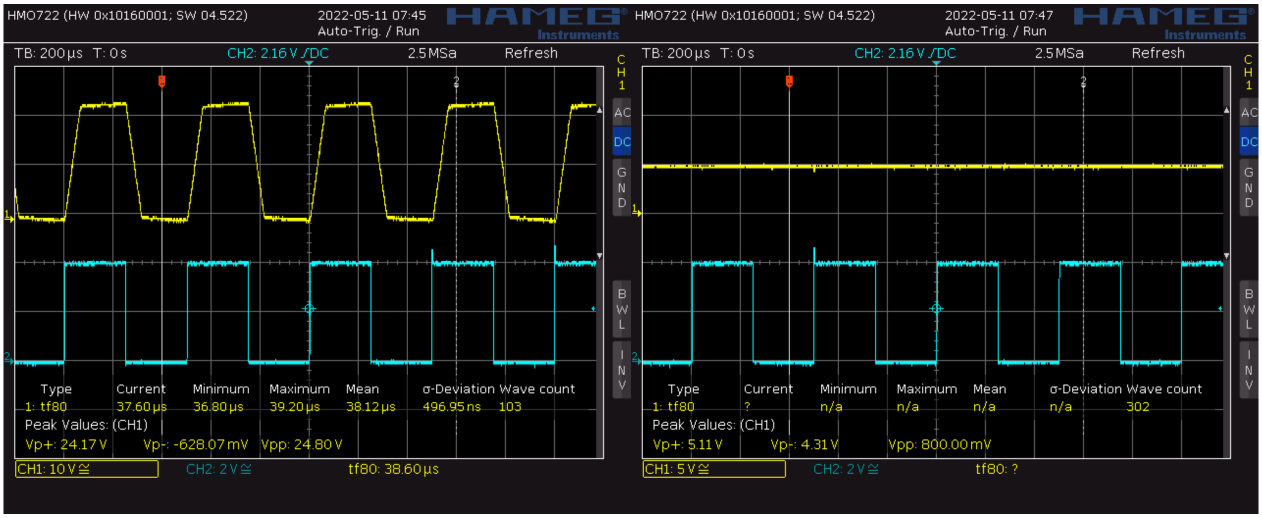


Figure 5.13: The circuits with simulated responses using a PWM signal with a 50% duty cycle. (a) The initial RC-filter. (b) Filter response of (a) with 24V PWM signal. (c) The testing circuit was used. (d) Circuit response of (c) when using a 4V PWM signal.



(a) (b)

Figure 5.14: The measurement of the signal from different points in the circuit. (a) In yellow the signal at V_s and in blue V_{in}. (b) In yellow the signal at V_1 and in blue V_{in}.

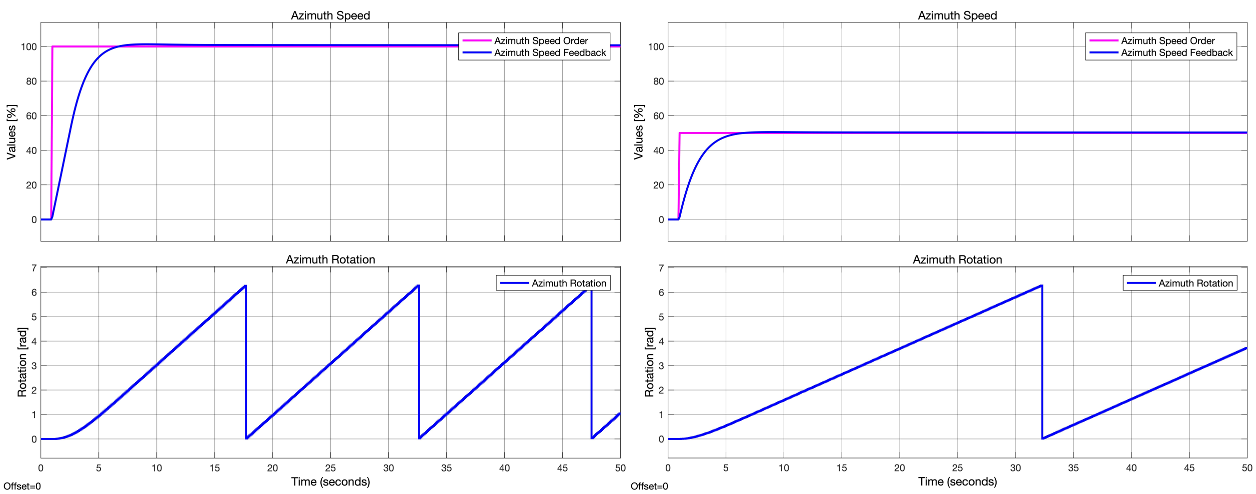
5.5.3.2 Model

From the factsheet, the total dryweigth of the thruster was selected to be 17 tonnes, with an assumed mass ratio of 0.4. This means that 40% of the weight is located at the lower part of the thruster D_1. Further, the radius of the propeller was selected to be one meter from the factsheet and the radius of the vertical cylinder D_2 was estimated to be 0.5 meters. The length of the lower part of the thruster (L_{D1}) was assumed to be three meters. The total moment of inertia was calculated to be 8075 from 4.7. For the inertia of the motor, it was selected to use 30, and the total inertia J_m was calculated to be 8105 from 4.8. The other parameters were selected to achieve the desired response. Hence, the motor parameters $0.8Nm/A$, $0.8Vs$ and 0.23Ω was selected for K_t, K_m and R_a, respectively, and the dampening coefficient D_m was selected to be zero. The complete transfer function can therefore be written as:

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{1.25}{2913s+1} \tag{5.3}$$

The model for the azimuth function was implemented as a continuous model and discretised with a sample time of $0.1s$, and the integrators were changed similar to the other models. There was also added a delay block to solve the algebraic loop for the feedback. The PID-controller used the parameters in Table 5.2 for the azimuth. When testing and comparing the continuous and discrete model, there was no notable difference, and therefore the results from the continuous model are omitted. In Figure 5.15a, the rotation for the azimuth function is relative to the speed order and feedback. The speed order is controlled by a step signal from 0% to 100%.

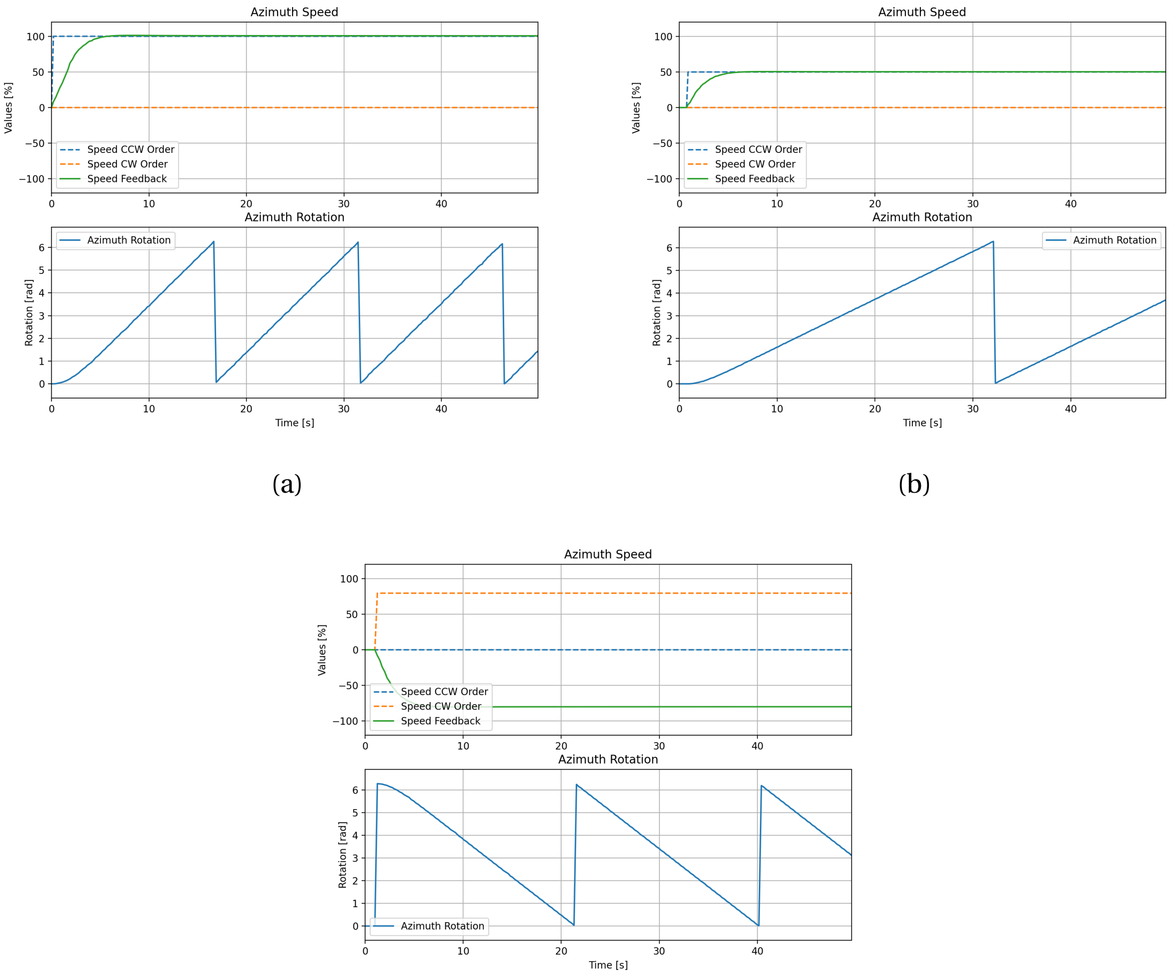
Notice that when the speed order is halved from 5.15b to 50%, the time for a complete rotation is doubled as expected.



(a) (b)

Figure 5.15: The azimuth position and relative rotation speed. (a) Going from 0% speed order to 100%. (b) Going from 0% speed order to 50%.

The implemented model was tested together with the circuit tested above. So the order is given directly from the function generator through the circuit pictured in Figure 5.4 and the output from the circuit is either connected to the CCW or the CW signal on the test jig. Therefore, this tests the complete control loop. The first two tests replicates the tests done on the simulated model with one step function to give an order of 100% in the CCW direction shown in Figure 5.16a. The second test in Figure 5.16b sets an order of 50% in the CCW direction. Both these tests replicate the response given by the simulated model. Finally, the CW direction was tested with a step signal to 80%. Notice how the speed is negative even though the order is positive in Figure 5.16c



(a) (b)

(c)

Figure 5.16: The azimuth position and relative rotation speed measured for the system where a positive speed means CCW rotation. (a) Going from 0% speed order to 100%. (b) Going from 0% speed order to 50%, going from 0% speed order to -80%.

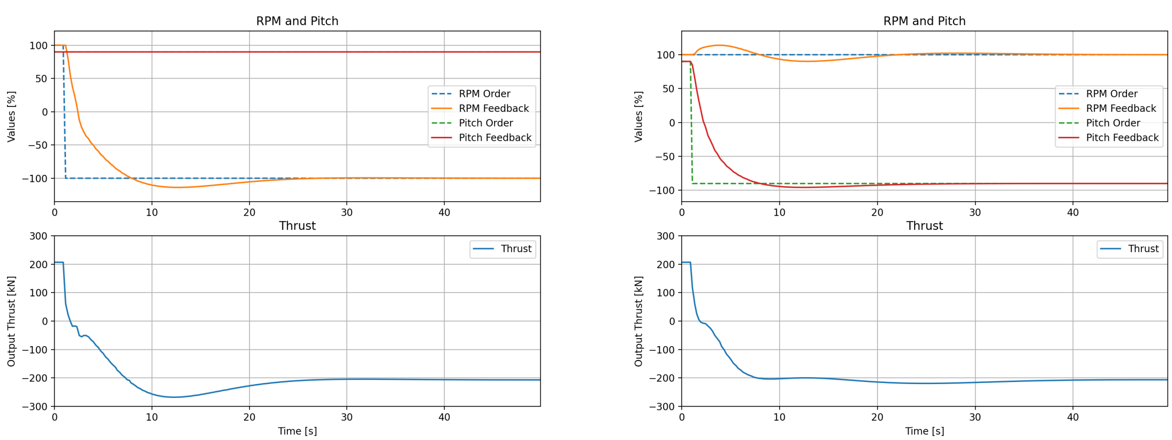
5.6 Extended testing of the propeller model

The extended testing on the implemented thruster simulator included how the system responds to a sudden change in thrust direction and how the model can follow a gradual change in the order value that affects the feedback.

5.6.1 Change of direction

There are two ways the propeller model can change thrust direction. The first way is to reverse the rotation direction of the propeller. In Figure 5.17a, the response of the thruster model is shown when the RPM order is set to -100% at the time 1s. Before the change, the propeller was set to an order of 100% and the feedback had stabilised at this value. In the test, the pitch order was set to 90%, and when the RPM was changed, there was no change in the pitch control. Notice how the response is not linear and the feedback response is faster when the thrust is closer to zero.

The second way the model can change thrust direction is to rotate the pitch of the propeller so the angle of attack is reversed. Figure 5.17b shows the response of the model when the pitch order and feedback were set to 90% and changed to -90% at the time of 1 second. Observe how the RPM feedback has an overshoot when the pitch is lowered toward zero and an undershoot when the pitch angle moves away from zero.



(a) (b)

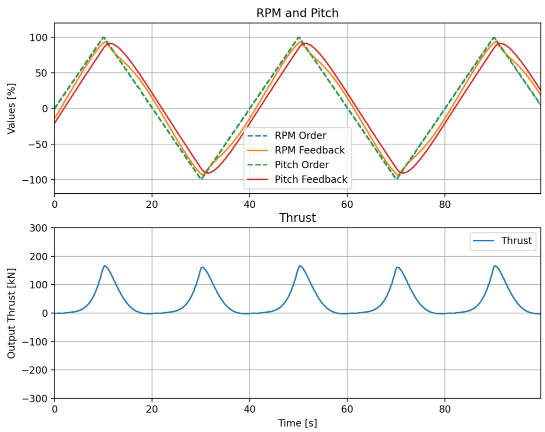
Figure 5.17: The system response for changing thrust direction. (a) Going from 100% RPM order to -100% with a constant pitch of 90%. (b) Going from 90% pitch order to -90% with a constant rpm of 100%.

5.6.2 Reference tracking

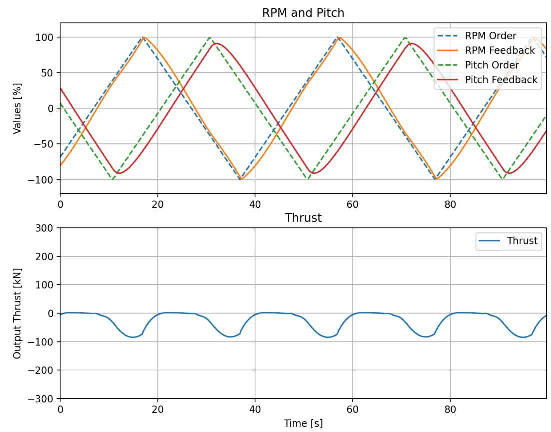
To test the reference tracking of the model, there was used a function generator to apply different signals to the pitch and RPM order. When applying the same triangular signal for both the pitch and RPM order with a period time of 20 seconds the response in Figure 5.18a was achieved. This response shows that the pitch feedback tracks the reference with a time delay, but there is a greater deviation right after the peaks and flattens the response. Observe that the thrust is always in the same direction even though the pitch and RPM changes direction.

In Figure 5.18b, the result from using the same triangular signal as above but with a phase shift between the rpm and pitch order. This was done by using two separate function generators set at the same signal parameters but started at different times to get the phase shift.

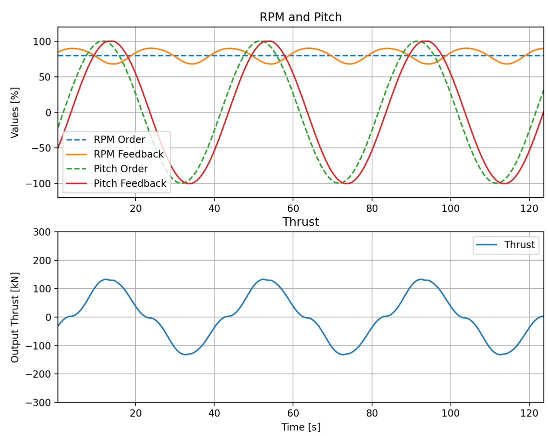
When testing the model by applying a sinusoidal signal to the pitch order, the response in Figure 5.18c was achieved. The RPM order was set to a static value of 80% and the sinusoidal signal had a period time of 20 seconds. Notice how the peaks and troughs in the pitch feedback are reflected in the RPM feedback with double the frequency. The thrust graph shows that there occur saddle points when the pitch is zero.



(a)



(b)



(c)

Figure 5.18: The response when applying different order signals to RPM and pitch. (a) When applying the same triangular signal with a period time of 20 seconds as the order for RPM and pitch. (b) When applying two out of phase triangular signals to the RPM and pitch order. (c) By applying a sinusoidal signal to the pitch order and an constant rpm order of 80%.

Chapter 6

Discussion

This chapter discusses and evaluates the results achieved in the project. First, the complete system is discussed, including the OPC implementation. Next, the development process in 4diac is discussed, where the main problems are evaluated with possible solutions and causes. Then the dynamic models are discussed, including the improved propeller model and the azimuth function with the RC-filter circuit. Followed by a discussion of the thruster simulator program made in 4diac, focusing on the implementation of the different parts. Next, how the project is limited and how the system can be further developed. This includes the test setup, model improvements and Finally, the project experiences and the group dynamic is discussed with a focus on the communication and planning of the project.

6.1 Complete system

From the gathered results it is shown that the complete system works. Functionality for two new thruster types were added to the system. The physical system was expanded to test the new functionalities. A signal generator and power supply was added to the physical system to simulate a PWM signal, and an RC circuit to convert the PWM signal to an analogue signal, to make it compatible with the PLC. This solution worked well. To test the rest of the new functionalities the test jig had to be used. Usually the test jig is used to simulate signals to Mcon, but in this case it was used to simulate Mcon. It was originally planned to test the new features with the actual Mcon but unfortunately the new software was not available during the project period so some improvisation was necessary. Using the test jig was a good solution for this situation, but it can only be determined if the simulator is completely compatible with Mcon after it is tested with Mcon.

The system successfully incorporates the OPC protocol, but the method uses two Forte instances running on the same controller. The main advantage of doing it this way is that it works without making large changes in the ptxdist compiler and Forte source. One of the disadvantages to the system is that there are two running versions of Forte where one is dedicated to only OPC communication. This overcomplicates any expansion of the OPC protocol. For example, if an extra value must be added to the OPC protocol, it must first be published from the first Forte version. Then the signal must be subscribed to before it again must be published as an OPC value. This also increases network traffic since the UDP packet for internal communication is also broadcasted from the PLC. Another advantage is that it is possible to run the Forte runtime for the OPC on another device and potentially integrate multiple "thruster simulators" on a single OPC server. Moving the Forte version containing the OPC program can easily be done by starting the docker container on a different device. The only constraint is that the current setup must use the arm architecture.

When comparing the average execution time for the OPC logger to the update frequency of the program, one can conclude that when logging values for only two nodes, all of the value changes get accounted for. However, when collecting data from five nodes, the execution time is greater than the update frequency causing some of the updates to get overwritten before they could be saved to the log. This does not have major impact on the results other than making it an approximation instead of an accurate result.

6.2 Developing with 4diac

Multiple challenges became became apparent when developing this project in 4diac. Some of these were listed in section 5.4.4, and these reflect the user experience of using the program. Since 4diac is in constant development, most of the problems mentioned should eventually be fixed.

One of the main problems was finding the correct procedure to address the I/O modules.

From the test that was done, several non-intuitive patterns occurred. The main pattern was that the addressing was node-based, and the same type of I/O (i.e. AI, DO, etc.) was addressed with the same suffix, the number before the period in the address (e.g. in the address *3.6*, it is the 3). For example, in test 6 in Table 5.1, all the DIs start with 3, the AIs start with 1. The suffix address seems to be based on the position of the first module of that type for the digital I/O, but for the analogue I/O, it is static and the inputs always start with 1 and the output with a 2. When the position of the modules is decided, some modules can be ignored, like the filter module, which is clearly shown in test 11 when the DI starts with 0 even though the first DI is in the second position. To further complicate the issue, there was a case where the AO modules stopped working in a certain configuration, see tests 7 and 10 in Table 5.1. However, the problem was somehow fixed by adding a different DO module immediately after the first DO module. This may have been caused by a problem with that specific type of module, but it clearly shows that it is unpredictable and that all I/O configurations have to be tested. When changing the configuration, it must be tested again since one module can affect others in a significant way.

Another problem, was that there is no debugger included in the 4diac IDE. It is possible to monitor a running program and therefore it is possible to identify if an error occurs in the main logic or one specific function block. Any error in a function block is hard to identify since it is not possible to check exactly what happens inside a function block. This means it is impossible to check whether there is only a missing link or a fundamental problem in the logic used inside the function block. This complicates the development significantly since there is no way to visualise and check what happens inside a function block.

One advantage of using 4diac is implementing the IEC61499 standard and the ability to create a system-wide program. This made it easy to understand the program and how the different parts interact. For this project, it was only used two *controllers*, but it was a great help to quickly see what controller a specific block belongs to. The potential problem that can occur is that there are two different places it is possible to place function blocks, and any blocks placed directly in the resource are not represented in the app. This can cause confusion, but it can also be used to create a more readable program. This was used for the blocks for the internal communication, and this significantly reduced the clutter and made a more readable app by removing about 40 blocks.

One last disadvantage with developing with 4diac is having to make a new Forte version for every change done to a function block. However, this was mainly solved with the script to automate the process, which significantly improved the workflow. This script simplified the process, and there was no need to remember how to do the different methods for making local and remote versions.

6.3 Dynamic thruster models

In general, the parameterisation of the models was done to achieve a specific response. This causes the models not to really represent any real system. It was decided to simplify the models in this way to focus on the finished response and functionality of the complete system. All parameters should be estimated based on real data or gathered from exact sources to achieve a realistic model. However, since the parameters change based on the thruster type and size, there is no way to make one general model to represent multiple thrusters realistically. One solution could have been to make one exact model first and reuse it as a representation until an exact model can be made.

In the following sections, the thruster models are discussed, compared and evaluated based on the theoretical models described in chapter 2 and implemented in Section 4.3.2. First, the propeller model is evaluated and compared to the old model. Then the azimuth model is discussed including the RC-filter.

6.3.1 Propeller model

By comparing the step responses for the RPM feedback of the new model in Figures 5.10c and 5.11c, it is clear that the response is highly dependent on the pitch angle. This is logical since when the pitch is low there is less fluid pushed by the propeller, therefore a lower force is needed to turn the propeller. This is also further collaborated on by how the RPM feedback changes in Figure 5.17b and speeds up when the pitch approaches zero. Because of this relationship, it was impossible to get a good control response for all the pitch positions. Therefore, it was decided first to adjust the pitch PID and then adjust the RPM PID based on the step response when both the pitch and RPM receive a step response. This was based on the assumption that the initial position for the thruster would be with zero in pitch and RPM. Therefore, the response is underdamped with a static low pitch angle and overdamped with a high pitch angle.

When comparing the responses when changing direction for the old model in Figure 5.9d and the new model in Figure 5.17a the differences are apparent. Where the old response is quick and the feedback goes from 100% to -40% in a little over a second the new model uses about 2 seconds for the same difference. It is also possible to identify that the fluid speed does have an effect on the RPM response since when inspecting the response of the new model, it is quickest when the RPM an thrust is approaching zero. This is when the propeller has the least resistance. The new model does have a significant overshoot in the RPM and thrust response which is not ideal, and probably could be

The main advantage of the new model is that it is possible to change the thrust direction by changing the pitch angle as well. When inspecting this response in Figure 5.17b it is identified that this generates a more stable thrust change. The overshoot on the thrust is mostly eliminated compared to the RPM response, and the thrust is mostly stable 7 seconds after the change order. This is also the normal way to reverse the thrust if a CPP thruster is used.

6.3.2 Azimuth

The response of the azimuth function is not modelled to be realistic but to offer a model that can be used to test the function and introduce some dynamics to the model. The current response can be used as a model for a small thruster with good manoeuvrability. For larger thrusters, the response should be slower due to all the mechanics used to rotate the thruster and the size of the thruster itself. This was remarked by Kongsberg Maritime, however it was too late in the implementation for it to be changed and should have been checked before.

The filter response for the PWM transformation shows a slight deviation in the simulated response. This deviation is probably from the uncertainty in the components used and the measurement current in the AO module. Comparing the op-amp's output voltage for the simulated system in Figure 5.13d and the measured response in Figure 5.14a shows a rise time in the circuit that can also cause some deviation. The signal is reformatted in the program which removes the deviation. Hence, why the deviation is not noticeable in the Azimuth order signal in Figure 5.15.

When simulating the PWM signal that Mcon outputs for azimuth rotation, a frequency of

2kHz was used. It is not certain whether the Mcon signal has the same frequency. From the transfer function, it is possible to predict the filter response for other frequencies. With a higher frequency, the frequency would be more damped. Therefore, the rise time to steady-state would be higher, and the peak to peak ripple voltage would be lower. The response would be changed for lower frequencies to have a faster rise time with a larger ripple. Therefore, this filter is probably sufficient for frequencies above 2kHz since the response time is not a priority since the azimuth model is significantly slower than the filter.

6.4 The thruster simulator in 4diac

The main purpose of the thruster simulator in 4diac is to simulate responses to Mcon as expected in a physical system. This includes dynamic responses based on orders and responding to digital signals in a certain way. For much of this logic, the expected responses to Mcon were available to base the simulator on. Therefore, the simulator behaves as expected, relating to starting and stopping the pump and thruster. This is also true for the simulation of the signals Mcon expects from the DP and Joystick systems. The orders for pitch, rpm, and azimuth can be given from DP or Joystick mode.

The function blocks for Joystick, DP, and visualisation do not contain any logic. They are only used to sort and gather the signals relating to the same systems. Therefore, they do not serve much purpose as of now, but they help sort the main program into categories and are therefore considered useful. If at some point it would be desirable to expand them with some logic, they are made compatible with that.

The project experienced some delays regarding software and hardware updates to change the Mcon test rig from a tunnel thruster system to a swing-up azimuthing system. Therefore, the signal lists for the new thrusters were available to see which signals were added, but not the logic to know exactly the order of the expected responses from the simulator to Mcon. This only, affected the signals relating to the swing-up functionality, which has twelve outputs that need to be activated in the correct order, based on six inputs from Mcon. In the simulator, the logic for this was based on the group's interpretation of the signal lists and a group member's experience with testing similar systems. Because of the unavailable Mcon software and hardware update, the new additions to the simulator could not be tested with the Mcon test rack. The test jig was used instead. It is therefore not certain that all the newly added parts of the program such as the swing-up functionality, is compatible with the actual Mcon software, but it is believed to be for the most part. If it is incorrect, it should not take much work to change it either.

The swing-up function block is made so that when lowering or lifting has begun, the thruster cannot go back to the previous stage, it must follow through. If the thruster is in the lower position and has been unlocked, it can be locked again. However, if the lifting has begun it must reach the upper position and be locked and unlocked before it can be lowered again. It was made based on the assumption that it is impossible to stop in the Mcon system once the lifting or lowering has begun.

6.5 Further work and limitations

The complete system is working and incorporates all the main parts outlined in the project plan. For some parts of the system, it became apparent that the implementation is not the most user-friendly and may cause problems with further expansion. There has not been added visualisation to the system, making it more complicated to be used for FAT. However, there should not be a problem since the OPC UA protocol is implemented with an extended variable set available.

Several models can be improved for more realistic and accurate simulations. The main parts that should be considered are to change the motor model, pitch model and better estimation of the model parameters. The motor model can be improved by changing the DC motor to the correct motor type, an AC or PM motor for the thrusters in the current system. There is no feedback from the fluid dynamics for the pitch model, which would impact the response. Therefore, the current pitch model is not accurate. Since the parameters mainly were selected to achieve a certain response and not necessarily to the correct value, these must be improved to get an accurate simulation.

Since the feedback is not implemented with CAN-bus, it is not possible to test this model directly with the Mcon system in the current state. It should be a priority to fix this for further expansion since it would not be possible to test the azimuth function accurately without it. There was proposed a solution for how it can be implemented.

There are two PWM signals in the signal list for the swing-up, the lift order and lower order. These signals have been simplified to digital signals in 4diac, but this should be changed before using the simulator in actual testing. Since the PLC is not compatible with PWM the signal should be converted to an analogue 0-10 VDC using an RC circuit, like the azimuth rotational signals are. Then one option is to alter the function block so that the timer is activated when the analogue signal is above a set value. Another more detailed option is to replace the digital signal and timer with a dynamic model that uses the lift/lower speed from the analogue signal to determine when the thruster is in place. The choice of solution should be made based on how detailed it needs to be for its purpose.

Since the program for semi-automatic FAT was not working, it could not be tested with our system. The problem was identified, but due to the lack of competence in the group for Java development, it was decided not to prioritise fixing it. If the program was working, there could be added support for this system with minor changes. This is because the new system uses different OPC UA names and IDs. Since the OPC implantation was tested, there should not arise any issues.

6.6 Project experiences

The group had good dynamics and communication throughout the project. The group members were able to express what they expected from each other and handle important decisions as a group. A wide range of interests and skills helped the group handle the various parts of the project. The project worked well when assigning individual group members to each task, and helping each other when needed.

The project mostly followed the plan made in the pre-project. Some tasks were done in parallel instead of sequentially, in part due to delays in the tasks. A few unplanned tasks were added to the plan. Some because of unforeseen issues that needed to be solved, and some because of misinterpretations during planning. It was decided to not complete all the planned tasks. This was both because it was decided the unplanned tasks were more important to complete, but also because some tasks were more time-consuming than anticipated. This was particularly regarding the tasks related to porting the old simulator and researching a new CPP model.

Chapter 7

Conclusions

Three main goals were set in the pre-project. Although not all planned tasks were completed, the main goals have still been achieved.

The first goal was to port the old simulator to the open-source platform Eclipse 4diac, and replace the B&R PLC with Wago. The related software 4diac Forte was successfully installed on the Wago PLC. A bash-script was made to streamline the build and import of the Forte to the PLC. After fixing the issues with the Java program for FAT it should be able to work with minor changes. Values are sent through OPC UA to a logger on a computer.

The second goal was to expand the simulator with more thruster types. The simulator was made compatible with Swing-up Azimuthing and Azipull thrusters to complete this goal. This included expanding the simulator to handle adjustable pitch and azimuth. Two new Simulink models were created, one for azimuth and the other a new CPP model for the propeller RPM.

Since a more accurate CPP model was made than the previous one, the third goal was also achieved, to improve the existing dynamic models. It was originally planned to replace the DC motor model with PM and AC motor models, but there was not enough time due to unforeseen issues.

The parts of the simulator that could be tested with Mcon for tunnel thrusters are compatible with Mcon. The features added to the simulator when expanding with new thruster types have not been tested with Mcon since it was unavailable. The simulator should be compatible with Mcon for the most part, but some alterations may be needed. The PWM signals from Mcon need to be converted to analogue 0-10 VDC, and a solution for this was successfully implemented. A solution was presented to produce feedback to Mcon over CANbus, but this could not be implemented in the project due to a shortage of both time and parts.

7.1 Further work

There are multiple things that could be added or improved in the project next, but below is a list of tasks that the project group would prioritise. The list is made from a viewpoint of learning outcome combined with client interest.

- Test the latest features of the simulator with the Mcon system
- Make the dynamic models more accurate (AC and PM motor)
- Add visualisation for reading values and triggering tests

Bibliography

[1] Skybrary, *Blade pitch*. [Online]. Available: https://skybrary.aero/articles/blade-pitch (visited on 05/19/2022).

[2] E. S. Skarbø, L. H. Sperre, M. O. Furmyr, and N. H. Mellum, "Mcon Thruster Simulator," NTNU, 2021.

[3] Wikipedia, *Maneuvering thruster*. [Online]. Available: https://en.wikipedia.org/wiki/Maneuvering_thruster (visited on 04/11/2022).

[4] Kongsberg Maritime, *Permanent magnet tunnel thruster*. [Online]. Available: https://www.kongsberg.com/maritime/products/propulsors-and-propulsion-systems/thrusters/tunnel-thrusters/permanent-magnet-tunnel-thruster-tt-pm/tt-pm-2000/ (visited on 04/25/2022).

[5] ——, *Swing-up azimuthing thruster*. [Online]. Available: https://www.kongsberg.com/no/maritime/products/propulsors-and-propulsion-systems/thrusters/azimuth-thrusters/swing-up-tcnsc-azimuthing-thruster/ (visited on 05/08/2022).

[6] ——, *Azipull thruster*. [Online]. Available: https://www.kongsberg.com/maritime/products/propulsors-and-propulsion-systems/thrusters/tunnel-thrusters/permanent-magnet-tunnel-thruster-tt-pm/tt-pm-2000/ (visited on 04/25/2022).

[7] N. S. Nise, *Control Systems Engineering*, 8th ed. Nashville, TN: John Wiley & Sons, May 2019.

[8] A. Croft and R. Davison, *Engineering Mathematics*, 5th ed. London, England: Pearson Education, Jun. 2017.

[9] J. D. Irwin, R. M. Nelms, and A. Patnaik, *Engineering Circuit Analysis*, en, 11th ed. Nashville, TN: John Wiley & Sons, Jun. 2015.

[10] A. J. Healey, S. Rock, S. Cody, D. Miles, and J. Brown, "Toward an improved understanding of thruster dynamics for underwater vehicles," *IEEE Journal of oceanic Engineering*, vol. 20, no. 4, pp. 354–361, 1995.

[11] J. P. Brown, "Four quadrant dynamic model of the auv ii thruster," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 1993.

[12] M. B. McLean, "Dynamic performance of small diameter tunnel thrusters," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 1991.

[13] S. E. Cody, "An experimental study of the response of small tunnel thrusters to triangular and square wave inputs," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 1992.

[14] D. Miles, D. Burton, M. Lee, and S. Rock, "Closed loop force control of underwater thrusters," *Techreport, Monterey Bay Aquarium Research Institute and Stanford University Aerospace Robotics Laboratory*, 1992.

[15] J. Adams, D. Burton, and M. Lee, "Dynamic characterization and control of thrusters for underwater vehicles," *Techreport, Monterey Bay Aquarium Research Institute and Stanford University Aerospace Robotics Laboratory*, 1991.

[16] International Electrotechnical Commission, *Understanding standards*. [Online]. Available: https://iec.ch/understanding-standards (visited on 04/19/2022).

[17] PLCopen, *Status iec 61131-3 standard*. [Online]. Available: https://plcopen.org/status-iec-61131-3-standard (visited on 04/19/2022).

[18] ——, *Iec 61131-3*. [Online]. Available: https://plcopen.org/iec-61131-3 (visited on 04/19/2022).

[19] A. Zoitl and T. Strasser, Eds., *Distributed control applications : guidelines, design patterns, and application examples with the IEC 61499*, ser. Industrial Information Technology. London, England: CRC Press, Jul. 2017, ISBN: 9781482259063,1271271281,1482259060.

[20] Eclipse Foundation, *What is eclipse 4diac?* [Online]. Available: https://www.eclipse.org/4diac/ (visited on 03/10/2022).

[21] Docker, *Docker overview | docker documentation.* [Online]. Available: https://docs.docker.com/get-started/overview/ (visited on 04/27/2022).

[22] ——, *Dockerfile reference.* [Online]. Available: https://docs.docker.com/engine/reference/builder/ (visited on 04/27/2022).

[23] ——, *Use containers to build,share and run your applications.* [Online]. Available: https://www.docker.com/resources/what-container/ (visited on 04/27/2022).

[24] M. Damm, S.-H. Leitner, and W. Mahnke, *OPC Unified Architecture*, 1st ed. Springer-Verlag Berlin Heidelberg, 2009, ISBN: 3540688986,9783540688983,3540688994,9783540688990.

[25] SSH Academy, *Ssh (secure shell) home page.* [Online]. Available: https://www.ssh.com/academy/ssh (visited on 04/13/2022).

[26] CAN in Automation, *Canopen - the standardized embedded network.* [Online]. Available: https://www.can-cia.org/canopen/ (visited on 04/26/2022).

[27] ——, *Canopen internal device architecture.* [Online]. Available: https://www.can-cia.org/can-knowledge/canopen/device-architecture/ (visited on 05/11/2022).

[28] Eclipse Foundation, *Building 4diac forte for wago.* [Online]. Available: https://www.eclipse.org/4diac/en_help.php?helppage=html/installation/wago.html (visited on 04/25/2022).

[29] WAGO Kontakttechnik GmbH & Co. KG, *Wago/pfc-firmware-sdk-g2: Firmware sdk for pfc200-g2 family.* [Online]. Available: https://github.com/WAGO/pfc-firmware-sdk-G2 (visited on 04/25/2022).

[30] Eclipse Foundation, *4diac: Downloads.* [Online]. Available: https://www.eclipse.org/4diac/en_dow.php (visited on 04/30/2022).

[31] B&R, *X20m2436.* [Online]. Available: https://download.br-automation.com/BRP4440000000000000 X20MM2436-ENG_V2.33.pdf?px-hash=3376d25f217232fd04b6821c9a0d96c1&px-time=1652997512 (visited on 05/12/2022).

[32] Eclipse Foundation, *4diac: Supported communication protocols*. [Online]. Available: https://www.eclipse.org/4diac/en_help.php?helppage=html/communication/communicationIndex.html (visited on 05/07/2022).

Appendices

A Preproject report

PRELIMINARY PROJECT

FOR BACHELOR THESIS

Kunnskap for en bedre verden

TITLE:
Improving the testing of the Mcon propulsion control system

CANDIDATES (SURNAME, NAME):

GODØ, ESPEN BRANDTZÆG
HARAM, MARI ELLINOR LONGVA
RAMSTAD, OLE-ANDRE AURVÅGLID

DATE:	SUBJECT CODE:	SUBJECT:	DOCUMENT ACCESS:
23/01/22	**IELEA2920**	**Bachelor Thesis Automation**	- Open

STUDY:	NR. PAGES/ATTACHMENTS:	BIBL. NR:
ELECTRICAL ENGINEERING	17/4	- Not used -

CLIENT/SUPERVISOR(S):

Kongsberg Maritime
Ottar L. Osen
Robin T. Bye

Summary:

Kongsberg Maritime wants to improve and expand upon an earlier developed simulator, which is used to test and verify Mcon thruster control systems without access to the physical thruster.
This task is given as a bachelor thesis to students at the department of ICT and Natural Sciences.

The proposed solution will consist of improving the already made control system which represents the thruster and expanding the arsenal of available thrusters which one can simulate. One of the points of improvement, is to remove the necessity for licensed software and convert to an open-source software. To improve the already made control model, it is decided to remove the simplifications and assumptions of the models.

CONTENTS

1 INTRODUCTION

The group came together through a common interest in both the Mcon Thruster Control System, and the application of automation within the maritime field. Most of the group have had previous interactions with Kongsberg and considered it a positive experience, which encouraged the group more to accept Kongsberg as their client.

Kongsberg Maritime is a world leader in marine technology and a major marine systems provider. They supply for a vast number of ships regardless of their work tasks, such as the navy, merchant ships and offshore [1].

The task given by Kongsberg Maritime is to expand upon a previously developed SIL simulator. The simulator is a bachelor thesis made by previous students at NTNU. The purpose of the simulator is to replace the requirement for physical hardware to test and verify the thruster controller.

The group has decided to expand the simulator by adding additional thrusters, improving the mathematical models for the simulator and replace the need for a licensed software. If the group's workload is considered too small, the group is prepared to tackle to visualise the simulator using the Morild interface.

2 TERMINOLOGY

Mcon – Kongsberg Maritimes control system for propulsion and thrusters

FAT – Factory Acceptance Test

SIL – Software In-The-Loop

CPP – Controllable Pitch Propellers

FPP – Fixed Pitch Propellers

NDA – Non-Disclosure Agreement

Morild Interface – Training simulator for maritime industry

PM – Permanent Magnet

FMU – Functional Mock-up Unit

PLC – Programmable Logic Controller

3 PROJECT ORGANISATION

3.1 Project group

Student numbers
Mari Ellinor Longva Haram, 522476 Ole-Andre Aurvåglid Ramstad, 494693 Espen Brandtzæg Godø, 517299

Table: Student numbers for everyone in the group that delivers this assignment for evaluation in the subject IELEA2920.

3.1.1 Tasks for the project group – organisation

The group decided that the roles as project leader and secretary should rotate between the group members, so that each member gets the opportunity to have extra responsibilities. Circulating will simultaneously ensure that no one person is obligated to spend more time on organisational tasks and therefore does not get to spend as much time working on the project itself as the rest of the group. It was decided that each group member should rotate between the roles twice, to avoid someone only being project leader in a period with little activity due to the mandatory subject Systems Engineering. The duration was therefore set to 14 days.

Time period	Project leader	Secretary	Project engineer
17.01-30.01	Ole	Espen	Mari
31.01-13.02	Espen	Mari	Ole
14.02-27.02	Mari	Ole	Espen
28.02-13.03	Ole	Espen	Mari
14.03-27.03	Espen	Mari	Ole
28.03-10.04	Mari	Ole	Espen
11.04-24.04	Ole	Espen	Mari
25.04-08.05	Espen	Mari	Ole
09.05-20.05	Mari	Ole	Espen

3.1.2 Tasks for project leader

The responsibilities of the project leader are as follows:

- Plan meetings, write meeting agenda, lead meetings
- Update Gantt chart
- Check that weekly hours are logged
- Follow up on tasks (delegation)

3.1.3 Tasks for secretary

The responsibilities of the secretary are as follows:

- Write meeting minutes, make meeting minutes available
- Progress report

3.1.4 Tasks for project engineer

The responsibilities of the project engineer are as follows:

- Follow up on use of the right tools (version control, etc.)
- Filing structure
- Complete assigned tasks

3.2 Steering group (mentors and contact person client)

Mentors from NTNU:

1. Ottar L. Osen
2. Robin T. Bye

Contact at Kongsberg Maritime:

- Håkon Lunheim
- Geir Olav Otterlei

4 AGREEMENTS

4.1 Agreements with client

There is no NDA or other written agreements between the project group and the client Kongsberg Maritime.

4.2 Workplace and resources

The workplace will be at a lab facility at NTNU Ålesund. The test rack and other necessary equipment will be placed there. For assistance from Kongsberg Maritime, the project group can contact Håkon Lunheim. Weekly progress reports produced in the internal meetings will be published on a file sharing platform that the steering group has access to. The project group will have meetings with the steering group every other week.

4.3 Group norms – cooperation – attitude

- Core time 9-15 Mon-Fri, ideal work time 8-16 Mon-Fri.
- If a group member cannot work during core hours, the other group members should be notified beforehand, and the absent group member should work in the lost hours.
- Whether the group members should work from home office or by the test rack should be decided throughout the project based on current rules and regulations related to the Covid-19 pandemic and the current tasks in the project.
- Work hours should be logged daily.
- The subject Systems Engineering is mandatory and should therefore be prioritised on the days of lecture, without the need to work in the hours not spent on the bachelor thesis.
- Meeting with steering group every other week.
- Internal meeting at the end of each week with a progress report.

In communication the group members should be patient and understanding. All group members should be able to voice their opinions respectfully, and all opinions are to be respected. If someone is stuck on a task, they should ask the rest of the group for help on how to proceed with it. The group members should all work to meet deadlines. As automation engineers the group should also consider if the solution at hand is the best, or if it can be improved further.

5 PROJECT DESCRIPTION

5.1 Problem statement – objective – purpose

When executing a FAT all the functionality of the Mcon control system is tested and verified. The test is done today by connecting a physical test jig to the system and the testing jig must have manual interaction to give feedback to the control system. In 2021 a system was developed to semi-automate this process and included thruster simulation with testing and verification of the control system. This system was developed mainly in a closed system using B&R controllers and software.

This project's objective is to extend the functionality of this system. The extended functionality is to open the platform, so it is not dependent on B&R, and develop new and improve the models used in the simulator.

4diac is an open-source alternative for programming and developing programs for PLCs. This alternative will be tested and researched with the main objective of being a replacement for the B&R system currently in use. Further the ability to use FMUs and how these can be implemented will be explored since a FMU is platform and software independent and is supported by many modelling and simulation applications.

There are multiple other thruster types that the system needs to be able to test if it should be a viable replacement for the current FAT-process. Two of these thruster types is the Swing-up azimuth thruster and Azipull-PM. Both are available with FPP and CPP and the latter use a PM motor.

To simplify the FAT a visual representation of the thruster from the simulation would help troubleshooting and get an overview of the problem. Hence, there is a stretch goal to implement a Morild interface to enable a real-time 3D visualisation of one thruster or all the thrusters controlled by the Mcon system.

5.2 Requirements for solution or project result – specification

The solution and system developed should use an open standard, use universal models, be scalable and is suitable for further development. Further the system must be able to produce repeatable results.

At the end of the project a complete report explaining the setup, functionality, and usage of the system. The report will include an analysis of the result with suggestions for further work.

5.3 Planned procedure(s) for the development work – method(s)

The complete requirements of the project are not known, and the project must adapt to change based on feedback received during the project. Hence, it was decided to use an agile project management process with some inspiration from guidelines described by S. H. Danielsen in [2]. A disadvantage of an agile model is that it is harder to plan the entire scope of the project. The project is divided into multiple phases where each phase is a mini project. For every phase there are several sub tasks that must be completed before the milestone is reached, each milestone is also the end of a feature. Every feature improves the system and is a working product, as recommended when using an agile workflow [3].

During the project the Gantt chart, FMEA and project report should be updated during the project as recommended in [2].

5.4 Information gathering – performed and planned

During the preliminary report the group has acquired some relevant literature regarding the client's thrusters, propulsion, and control systems. The group has been provided with the thesis for the simulator, which is a great source of information on topics such as how it functions, how it was developed and where one could look for information. Information regarding the planned software 4diac is open source and there is documentation for how to use the software on their website.

5.5 Assessment – analysis of risk

The group considers the goals set to be achievable, with sufficient effort and research. Should the task of implementing two thruster types be too overwhelming, the group will settle with completing one and improving the already existing thruster configuration.

To make the goals achievable the group wants to emphasize communication and keeping the goals realistic.

In Appendix B the different risks of the project are assessed. In summary the main risks for the project are due to the ongoing pandemic and the delays this can cause. The actions taken to mitigate the risk it was decided to not have physical meetings unless necessary and use the lab as much as possible when it is available. Another action taken is to use version control and cloud storage to mitigate the risk of losing progress in the project from either human error (e.g., accidentally deletion) or computer failure.

5.6 Main activities in further work

The following activities and milestones are defined in this project.

Task Nr.	Task	Accountable	Time
A1	4diac simulator	Espen	33d
A11	Research basic and assess	Mari	7d
A12	Research communication	Ole-Andre	2d
A13	Test setup	Espen	7d
A14	Complete feature list of old system, planning of porting	Mari	2d
A15	Port old system	Ole-Andre	14d
A16	Verify that FAT works	Espen	2d
A17	Complete system in 4diac (Milestone)		
A2	Multiple thruster models	Mari	28d
A21	Research thrusters	Ole-Andre	7d
A22	Sketch thruster models, plan the making	Espen	2d
A23	Make model thruster typ1	Mari	14d
A24	Test thruster typ1	Ole-Andre	5d
A25	Make model thruster typ2	Espen	14d
A26	Test thruster typ2	Mari	5d
A27	Complete multiple thrusters (Milestone)		
A3	Improve models	Ole-Andre	28d
A31	AC motor model	Espen	14d
A32	PM motor model	Mari	14d
A33	FPP models	Ole-Andre	14d
A34	CPP models	Espen	14d
A35	Complete improvement (Milestone)		

A4	Visualization (stretch goal)	Espen	N/A
A41	Research software	Mari	2d
A42	Make concept draft	Ole-Andre	7d
A43	Complete version	Espen	14d
A44	Testing and QC	Mari	2d
A46	Complete Visualization (Milestone)		
A5	Thesis report	Espen	117d
A51	Draft A1	Mari	7d
A52	Draft A2	Ole-Andre	14d
A53	Draft A3	Espen	20d
A54	Complete Report	Mari	14d
A55	Complete project (Milestone)		

5.7 Progress plan – management of the project

5.7.1 Master plan

The first main activity A1 consist of evaluating whether it is possible to port the Mcon simulator to an open-source alternative, namely 4diac by Eclipse. This will consist of first doing research about the capabilities and functionality of 4diac. If it is decided that 4diac is a viable option, the system should be ported over.

Porting of the system is a substantial task and must be thoroughly planned based on the research done in the previous tasks. The structure will be to implement one feature then test and verify the functionality, then the next feature can be implemented.

The second main activity A2 is to develop two new thruster models for the system. This includes doing research of how they work, the physical model of the thrusters and planning how a simplified version can be implemented in the system. Further the models must be made according to the developed plan before it can be tested and verified.

For the third main activity A3 the goal is to further improve all the models for a more realistic response from the models. This includes making a new model for an AC motor and PM motor, improve the CPP and FPP models for the thrusters. Each of these activities must be divided and clarified and this should be done as part of creating the simplified models in A2.

As a stretch goal, the fourth activity A4 is adding a visualization. This could be a 3D view of the propeller or thruster, showing how it moves when changing inputs like RPM, pitch, and angle. In that case it would be based on a solution from Morild Interaktiv.

The last main activity A5 is to write the report documenting the thesis. The report should describe the results achieved in the project, and detail the theory and methods used. Parts of the report should be written alongside the work on the project while the experiences are fresh. After finishing the other activities in the project there should be a period where the report is the main focus.

5.7.2 Management aids

The Gantt chart in Appendix A is a complete overview of the activities and milestones described in section 5.6. Further, Appendix A also include the template for the hour list and a more in-depth task overview.

During the project the Gantt chart and the task overview must be updated when changes and deviations occurs. The hours used by each member must also be added to the hour list as described in section 4.3.

5.7.3 Developments tools

For this project the following tools will be used:

4diac – Programming environment

B&R Automation Studio – Programming and testing of old setup

MATLAB / Simulink – simulating and design of control models

Morild Interaktiv – For visualization

5.7.4 Internal control – evaluation

At the weekly meetings with the project group, and the biweekly meetings with the steering group, progress will be discussed and logged. The group members should have contact regularly and can also address any delay in the progress outside of the meetings, to resolve it sooner if possible.

Smaller goals are seen as achieved when tests prove the solution to be able to complete its purpose, and the task is executed good enough to avoid having to redo the task later. It is important to differentiate between completed and perfect. In addition to the group member responsible for the task, another group member should give it a quick look and agree that the goal is complete. For bigger goals the steering group will have to approve before it can be marked complete.

5.8 Decisions – decision process

In this pre-project all the decisions were made and agreed upon by the complete group. The activities and the agreements with Kongsberg Maritime were agreed upon by both the project group and the steering group, this includes the project objective and requirements.

Every major decision made by the group is made is made in unity. If there is a dispute a compromise where compromise where everyone agrees is tried to be reached. The decision is made by majority vote (decision that gets at least two votes) if no compromise can be found.

6 DOCUMENTATION

6.1 Reports and technical documents

At the end of each week during/after project group meetings, a progress report will be produced, see chapter 7.2.1 for more detail. Meeting minutes will be written by the secretary for each meeting and should be written the same day. All these documents, including necessary technical data from Kongsberg Maritime, will be stored on a filesharing platform that the steering group has access to.

7 PLANNED MEETINGS AND REPORTS

7.1 Meetings

7.1.1 Meetings with the steering group

The project group will meet with the steering group every other week. The project leader will call in to this meeting. The content of the meeting will be discussing progress and recent results, as well as the next steps moving forward. A summation of any problems encountered and how they were solved can also be given.

7.1.2 Project meetings

Meetings with just the project group will be held at the end of each week. The purpose of these meetings will be to sum up the achievements and focus areas of the week and prepare the list of tasks for the coming week. This meeting will also create the base for the weekly progress report.

7.2 Periodic reports

7.2.1 Progress reports (incl. milestones)

The project group will produce a progress report at the end of each week. The progress report will serve as documentation of the tasks worked on each week as well as how the project is progressing compared to the plan. The progress reports will be made available to the steering group through a filesharing platform, to keep them updated and to serve as a base for the next meeting with the steering group.

8 PLANNED DEVIATION TREATMENT

When a deviation in the execution of the tasks or agreements is discovered, the effected parties must be notified, and the impact analysed. The root cause of every deviation must be identified and if applicable new routines should be implemented to further to ensure that it is not repeated.

For minor deviations it is sufficient to only notify the accountable of the task with the issue and the resolution. A minor deviation is defined as a deviation that only cause a slight time delay and have no repercussion for the completion of the task.

If the deviation causes a delay for completing the task or if the deviation can impact other tasks it is defined as an intermediate deviation. Intermediate deviations must be processed by at least two group members where one of them is the member accountable of the task.

If a deviation significantly impacts the progress of the project or causes a significant delay for completing the project, it is defined as major deviation. Major deviations must be processed by the complete project group and the steering group must be informed in the next progress meeting.

9 EQUIPMENT REQUIREMENTS

- Testing equipment (Kongsberg Maritime)
- MATLAB / Simulink (NTNU license)
- B&R Automation Studio (Kongsberg Maritime)

10 BIBLIOGRAPHY

[1] Kongsberg, "Who We Are," 19 1 2022. [Online]. Available: https://www.kongsberg.com/no/maritime/about-us/who-we-are-kongsberg-maritime/.

[2] S. H. Danielsen, Fremragende forbedringsprosjekter med Lean Six Sigma, Drammen: Aksena Press AS, 2016.

[3] R. K. Wysocki, Effective Project Management: Traditional, Agile, Extreme, Hybrid, Indianapolis: John Wiley & Sons, Inc, 2019.

APPENDIX

Improving the testing of the Mcon propulsion control system

NTNU / Kongsberg Maritime

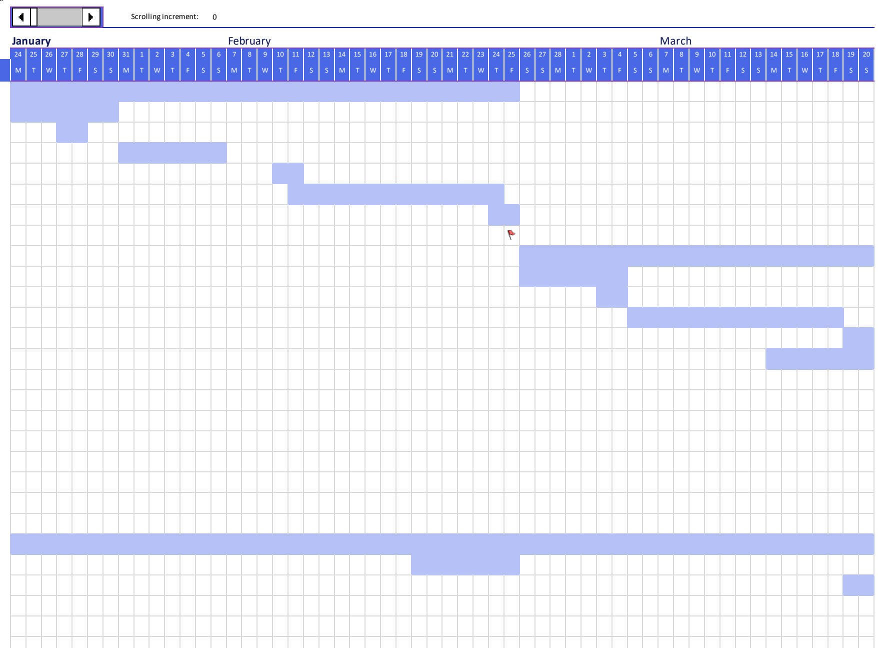
Espen Brandtzæg Godø, Mari Longva Haram, Ole-Andre Aurvåglid Ramstad

Project start date: 24/01/2022

Milestone marker: 1

Milestone description	Assigned to	Progress	Start	Days
A1 - 4diac simulator	Espen		24/01/2022	33
A11 - Research basic and access	Mari	0%	24/01/2022	7
A12 - Research communication	Ole-Andre	0%	27/01/2022	2
A13 - Test setup	Espen	0%	31/01/2022	7
A14 - Complete feature list of old system, planning of porting	Mari	0%	10/02/2022	2
A15 - Port old system	Ole-Andre	0%	11/02/2022	14
A16 - Verify that FAT works	Espen	0%	24/02/2022	2
A17 - Complete system in 4diac (Milestone)			25/02/2022	0
A2 - Multiple thruster models	Mari	0%	26/02/2022	28
A21 - Research thrusters	Ole-Andre	0%	26/02/2022	7
A22 - Sketch thruster models, plan the making	Espen	0%	03/03/2022	2
A23 - Make model thruster typ1	Mari	0%	05/03/2022	14
A24 - Test thruster typ1	Ole-Andre	0%	19/03/2022	5
A25 - Make model thruster typ2	Espen	0%	14/03/2022	14
A26 - Test thruster typ2	Mari	0%	28/03/2022	5
A27 - Complete multiple thrusters (Milestone)			01/04/2022	0
A3 - Improve models	Ole-Andre		02/04/2022	28
A31 - AC motor model	Espen	0%	02/04/2022	14
A32 - PM motor model	Mari	0%	02/04/2022	14
A33 - FPP models	Ole-Andre		16/04/2022	14
A34 - CPP models	Espen	0%	16/04/2022	14
A35 - Complete improvement (Milestone)		0%	29/04/2022	0
A5 - Thesis report	Espen	0%	24/01/2022	117
A51 - Draft A1	Mari		19/02/2022	7
A52 - Draft A2	Ole-Andre		19/03/2022	14
A53 - Draft A3	Espen		10/04/2022	20
A54 - Complete report	Mari		06/05/2022	14
A55 - Complete project (Milestone)			20/05/2022	0

To add more data, Insert new rows ABOVE this one



Improving the testing of the Mcon propulsion control system

NTNU / Kongsberg Maritime

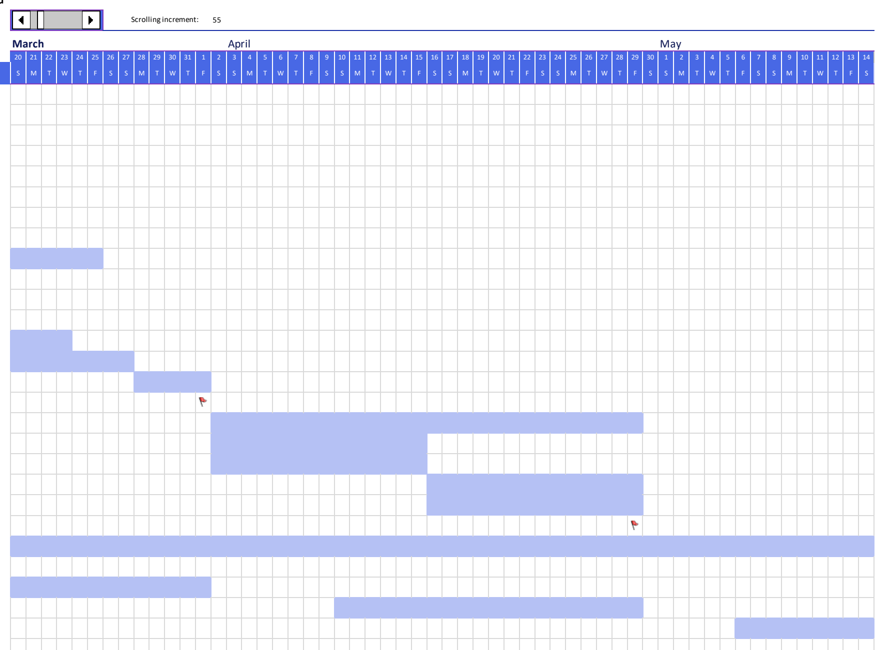
Espen Brandtzæg Godø, Mari Longva Haram, Ole-Andre Aurvåglid Ramstad

Project start date: 24/01/2022

Milestone marker: 1

Milestone description	Assigned to	Progress	Start	Days
A1 - 4diac simulator	Espen		24/01/2022	33
A11 - Research basic and access	Mari	0%	24/01/2022	7
A12 - Research communication	Ole-Andre	0%	27/01/2022	2
A13 - Test setup	Espen	0%	31/01/2022	7
A14 - Complete feature list of old system, planning of porting	Mari	0%	10/02/2022	2
A15 - Port old system	Ole-Andre	0%	11/02/2022	14
A16 - Verify that FAT works	Espen	0%	24/02/2022	2
A17 - Complete system in 4diac (Milestone)			25/02/2022	0
A2 - Multiple thruster models	Mari	0%	26/02/2022	28
A21 - Research thrusters	Ole-Andre	0%	26/02/2022	7
A22 - Sketch thruster models, plan the making	Espen	0%	03/03/2022	2
A23 - Make model thruster typ1	Mari	0%	05/03/2022	14
A24 - Test thruster typ1	Ole-Andre	0%	19/03/2022	5
A25 - Make model thruster typ2	Espen	0%	14/03/2022	14
A26 - Test thruster typ2	Mari	0%	28/03/2022	5
A27 - Complete multiple thrusters (Milestone)			01/04/2022	0
A3 - Improve models	Ole-Andre		02/04/2022	28
A31 - AC motor model	Espen	0%	02/04/2022	14
A32 - PM motor model	Mari	0%	02/04/2022	14
A33 - FPP models	Ole-Andre		16/04/2022	14
A34 - CPP models	Espen	0%	16/04/2022	14
A35 - Complete improvement (Milestone)		0%	29/04/2022	0
A5 - Thesis report	Espen	0%	24/01/2022	117
A51 - Draft A1	Mari		19/02/2022	7
A52 - Draft A2	Ole-Andre		19/03/2022	14
A53 - Draft A3	Espen		10/04/2022	20
A54 - Complete report	Mari		06/05/2022	14
A55 - Complete project (Milestone)			20/05/2022	0

To add more data, Insert new rows ABOVE this one



Task Overview

Phase	Task ID	Name	Subtask To	Assigned	Hours Accumulated	Start Date	Estimated Days	Date Complete	Description
	A01	Administration			0				
	A02	Meeting			10				
	A03	Miscellaneous			0				Use for all other tasks/hours used on the project
1	A1	4diac simulator		Espen	0	24/01/2022	33		Creating the simulator platform in 4diac
1	A11	Research basic and assess	A1	Mari	0	24/01/2022	7		Research the functionality of 4diac. IDE, FMU,
1	A12	Research communication	A1	Ole-Andre	0	27/01/2022	2		Check what fieldbus that can be used and how
1	A13	Test setup	A1	Espen	0	31/01/2022	7		Create a test project to check if the functionality is done based on the research
	A14	Complete feature list of old system, planning of porting	A1	Mari	0	10/02/2022	2		Make a complete overview of the functionality and plan how it can be iteratively implemented in the new system
1	A15	Port old system	A1	Ole-Andre	0	11/02/2022	14		system. Add one feature then test and verify before adding the next.
1	A16	Verify that FAT works	A1	Espen	0	24/02/2022	2		Complete a Factory acceptance test, using the new system. Check if the functionality is the same
2	A2	Multiple thruster models		Mari	0	26/02/2022	28		Add two new thruster models to the simulator
2	A21	Research thrusters	A2	Ole-Andre	0	26/02/2022	7		Research how the thrusters work, how they can be modelled, any simplifications that can be done
2	A22	Sketch thruster models, plan the making	A2	Espen	0	03/03/2022	2		Make a plan for the development process and divide and clarify the requirements.
2	A23	Make model thruster typ1	A2	Mari	0	05/03/2022	14		
2	A24	Test thruster typ1	A2	Ole-Andre	0	19/03/2022	5		Test and verify that the model works as expected, test with IO
2	A25	Make model thruster typ2	A2	Espen	0	14/03/2022	14		
2	A26	Test thruster typ2	A2	Mari	0	28/03/2022	5		Test and verify that the model works as expected, test with IO
3	A3	Improve models		Ole-Andre	0	02/04/2022	28		Improve the models used in the simulator
3	A31	AC motor model	A3	Espen	0	02/04/2022	14		Make new and improved model for a AC motor
3	A32	PM motor model	A3	Mari	0	02/04/2022	14		Make new and improved model for a PM motor
3	A33	FPP models	A3	Ole-Andre	0	16/04/2022	14		Make new and improved model for FP propeller
3	A34	CPP models	A3	Espen	0	16/04/2022	14		Make new and improved model for CP propeller
5	A5	Thesis report		Espen	0	24/01/2022	117		
5	A51	Draft A1	A5	Mari	0	19/02/2022	7		Add the info from phase 1 to the report
5	A52	Draft A2	A5	Ole-Andre	0	19/03/2022	14		Add the info from phase 2 to the report
5	A53	Draft A3	A5	Espen	0	10/04/2022	20		Add the info from phase 3 to the report
5	A54	Complete Report	A5	Mari	0	06/05/2022	14		Finalize the report

Hour list

Task	Date	Name	Hours	Comment
A02	18/01/2022	Ole	3	Meeting @NMK with KM regarding thesis
A02	14/01/2022	Ole	3	Meeting with group regarding prelimary report
A02	13/01/2022	Ole	4	Meeting with group regarding prelimary report

Failure mode and effects analysis

											Action Results				
Step	Potential Failure Mode	Potential Failure Effects	SEV	Potential Causes	OCC	Current Controls	DET	RPN	Actions Recommended	Responsibility	Action Taken	SEV	OCC	DET	RPN
	Shutdown due to covid-19	Unable to access the lab	4	Government, local restrictions	3		2	24	Use the lab as much as possible when it's available, try to follow the covid trends to detect any potential lockdowns	All members	Yes	2	3	2	12
	Sickness due to covid	Unable to work, transmitting to other group members	4	Infected from school, work, daily life	2	Local and national restrictions based on infection pressure	1	8	Avoid physical meetings unless necessary, stay at home if you have symptoms	All members	Yes	3	2	1	6
	Deviation from safety regulations	Physical injury, electrical shock, replacement of equipment	5	Faulty equipment, human error	1	Safety regulations	2	10				5	1	2	10
	Computer problems	Loss of data/progress	5	Computer failure, human error	2		2	20	Use version control and cloud storage for all project files	All members	Yes	1	2	2	4
	Time problem / not enough time	Delay of the project	3	Unexpected delays, changes to the requirements, time estimation errors, missing equipment	4		2	24	Weekly progress reports, bi-weekly meetings	All members	Yes	3	2	2	12
	Delivery issues	Delay of the project	4	chip-shortage, unable to access licenses	3		2	24	Request equipment as soon as possible	All members	Yes	2	3	2	12

B Gantt diagram with hour list

Improving the testing of the Mcon propulsion control system

NTNU / Kongsberg Maritime

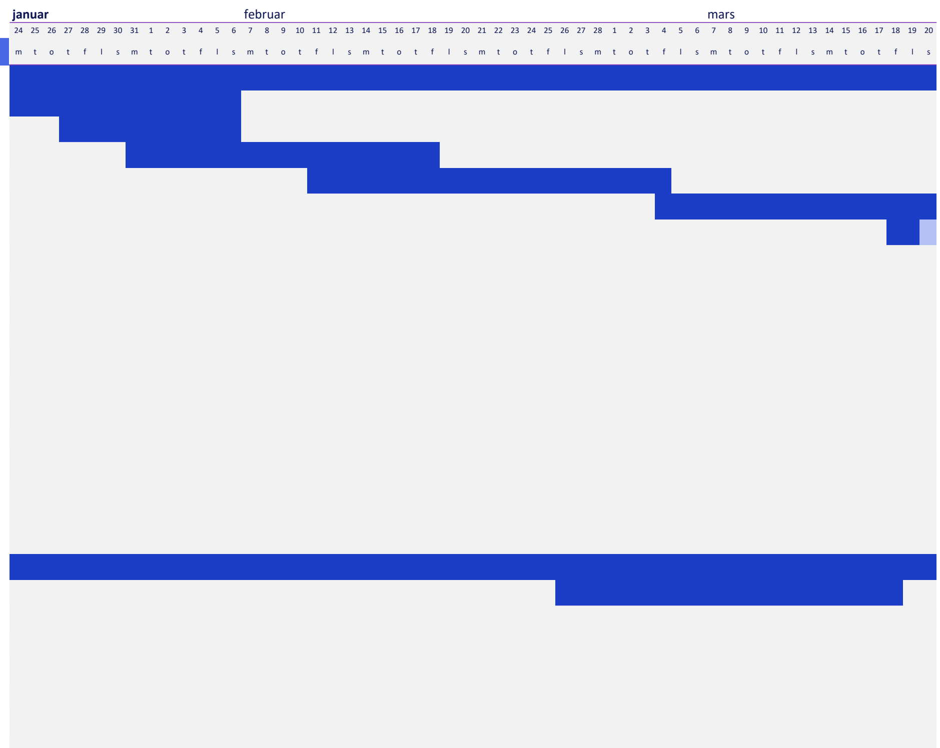
Espen Brandtzæg Godø, Mari Longva Haram, Ole-Andre Aurvåglid Ramstad

Project start date: 24.01.2022

Scrolling increment: 0

Milestone marker: 1

Milestone description	Assigned to	Progress	Start	Days
A1 - 4diac simulator	Espen	100 %	24.01.2022	68
A11 - Research basic and access	Mari, Espen, Ole-Andre	100 %	24.01.2022	14
A12 - Research communication	Ole-Andre, Mari, Espen	100 %	27.01.2022	11
A13 - Test setup	Espen, Mari, Ole-Andre	100 %	31.01.2022	19
A14 - Complete feature list of old system, planning of porting	Mari, Ole-Andre	100 %	11.02.2022	22
A15 - Port old system	Ole-Andre, Mari, Espen	100 %	04.03.2022	29
A16 - Verify that FAT works	Espen, Mari	20 %	18.03.2022	14
A17 - Complete system in 4diac (Milestone)		100 %	01.04.2022	0
A2 - Multiple thruster models	Mari	100 %	29.03.2022	44
A21 - Research thrusters	Ole-Andre, Mari, Espen	100 %	29.03.2022	14
A22 - Sketch thruster models, plan the making	Espen, Mari	100 %	31.03.2022	6
A23 - Make model thruster typ1	Mari, Espen, Ole-Andre	100 %	04.04.2022	37
A24 - Test and gather results th	Ole-Andre, Mari	100 %	03.05.2022	8
A25 - Make model thruster typ2	Espen, Ole-Andre	100 %	18.04.2022	22
A26 - Test and gather results thruster typ2	Mari, Espen	100 %	07.05.2022	5
A28 - OPC communication and logger	Ole-Andre, Espen	100 %	04.04.2022	25
A29 - Azimuthing function	Espen	100 %	25.04.2022	12
A27 - Complete multiple thrusters (Milestone)			11.05.2022	0
A3 - Improve models	Ole-Andre			10
A5 - Thesis report	Espen	100 %	24.01.2022	117
A51 - Draft A1	Mari	100 %	26.02.2022	21
A52 - Draft A2	Ole-Andre	100 %	04.04.2022	20
A53 - Draft A3	Espen	100 %	24.04.2022	14
A54 - Complete report	Mari	100 %	06.05.2022	14
A56 - Make presentation	Espen	100 %	13.05.2022	4
A57 - Make poster	Mari, Ole-Andre	100 %	16.05.2022	1
A55 - Complete project (Milestone)			20.05.2022	0



Improving the testing of the Mcon propulsion control system

NTNU / Kongsberg Maritime

Espen Brandtzæg Godø, Mari Longva Haram, Ole-Andre Aurvåglid Ramstad

Project start date: 24.01.2022

Scrolling increment: 55

Milestone marker: 1

Milestone description	Assigned to	Progress	Start	Days
A1 - 4diac simulator	Espen	100 %	24.01.2022	68
A11 - Research basic and access	Mari, Espen, Ole-Andre	100 %	24.01.2022	14
A12 - Research communication	Ole-Andre, Mari, Espen	100 %	27.01.2022	11
A13 - Test setup	Espen, Mari, Ole-Andre	100 %	31.01.2022	19
A14 - Complete feature list of old system, planning of porting	Mari, Ole-Andre	100 %	11.02.2022	22
A15 - Port old system	Ole-Andre, Mari, Espen	100 %	04.03.2022	29
A16 - Verify that FAT works	Espen, Mari	20 %	18.03.2022	14
A17 - Complete system in 4diac (Milestone)		100 %	01.04.2022	0
A2 - Multiple thruster models	Mari	100 %	29.03.2022	44
A21 - Research thrusters	Ole-Andre, Mari, Espen	100 %	29.03.2022	14
A22 - Sketch thruster models, plan the making	Espen, Mari	100 %	31.03.2022	6
A23 - Make model thruster typ1	Mari, Espen, Ole-Andre	100 %	04.04.2022	37
A24 - Test and gather results th	Ole-Andre, Mari	100 %	03.05.2022	8
A25 - Make model thruster typ2	Espen, Ole-Andre	100 %	18.04.2022	22
A26 - Test and gather results thruster typ2	Mari, Espen	100 %	07.05.2022	5
A28 - OPC communication and logger	Ole-Andre, Espen	100 %	04.04.2022	25
A29 - Azimuthing function	Espen	100 %	25.04.2022	12
A27 - Complete multiple thrusters (Milestone)			11.05.2022	0
A3 - Improve models	Ole-Andre			10
A5 - Thesis report	Espen	100 %	24.01.2022	117
A51 - Draft A1	Mari	100 %	26.02.2022	21
A52 - Draft A2	Ole-Andre	100 %	04.04.2022	20
A53 - Draft A3	Espen	100 %	24.04.2022	14
A54 - Complete report	Mari	100 %	06.05.2022	14
A56 - Make presentation	Espen	100 %	13.05.2022	4
A57 - Make poster	Mari, Ole-Andre	100 %	16.05.2022	1
A55 - Complete project (Milestone)			20.05.2022	0

Improving the testing of the Mcon propulsion control system

NTNU / Kongsberg Maritime

Espen Brandtzæg Godø, Mari Longva Haram, Ole-Andre Aurvåglid Ramstad

Project start date: 24.01.2022

Milestone marker: 1

Scrolling increment: 61

Milestone description	Assigned to	Progress	Start	Days
A1 - 4diac simulator	Espen	100 %	24.01.2022	68
A11 - Research basic and access	Mari, Espen, Ole-Andre	100 %	24.01.2022	14
A12 - Research communication	Ole-Andre, Mari, Espen	100 %	27.01.2022	11
A13 - Test setup	Espen, Mari, Ole-Andre	100 %	31.01.2022	19
A14 - Complete feature list of old system, planning of porting	Mari, Ole-Andre	100 %	11.02.2022	22
A15 - Port old system	Ole-Andre, Mari, Espen	100 %	04.03.2022	29
A16 - Verify that FAT works	Espen, Mari	20 %	18.03.2022	14
A17 - Complete system in 4diac (Milestone)		100 %	01.04.2022	0
A2 - Multiple thruster models	Mari	100 %	29.03.2022	44
A21 - Research thrusters	Ole-Andre, Mari, Espen	100 %	29.03.2022	14
A22 - Sketch thruster models, plan the making	Espen, Mari	100 %	31.03.2022	6
A23 - Make model thruster typ1	Mari, Espen, Ole-Andre	100 %	04.04.2022	37
A24 - Test and gather results th	Ole-Andre, Mari	100 %	03.05.2022	8
A25 - Make model thruster typ2	Espen, Ole-Andre	100 %	18.04.2022	22
A26 - Test and gather results thruster typ2	Mari, Espen	100 %	07.05.2022	5
A28 - OPC communication and logger	Ole-Andre, Espen	100 %	04.04.2022	25
A29 - Azimuthing function	Espen	100 %	25.04.2022	12
A27 - Complete multiple thrusters (Milestone)			11.05.2022	0
A3 - Improve models	Ole-Andre			10
A5 - Thesis report	Espen	100 %	24.01.2022	117
A51 - Draft A1	Mari	100 %	26.02.2022	21
A52 - Draft A2	Ole-Andre	100 %	04.04.2022	20
A53 - Draft A3	Espen	100 %	24.04.2022	14
A54 - Complete report	Mari	100 %	06.05.2022	14
A56 - Make presentation	Espen	100 %	13.05.2022	4
A57 - Make poster	Mari, Ole-Andre	100 %	16.05.2022	1
A55 - Complete project (Milestone)			20.05.2022	0

Task Overview

Phase	Task ID	Name	Subtask To	Assigned	Hours Accumulated	Start Date	Estimated Days	Date Complete	Description
	A01	Administration			6,5				
	A02	Meeting			57				
	A03	Miscellaneous			25,5				Use for all other tasks/hours used on the project
1	A1	4diac simulator		Espen	467,5	24.01.2022	33		Creating the simulator platform in 4diac
1	A11	Research basic and assess	A1	Mari	31,5	24.01.2022	7		Research the functionality of 4diac. IDE, FMU,
1	A12	Research communication	A1	Ole-Andre	2,5	27.01.2022	2		Check what fieldbus that can be used and how
1	A13	Test setup	A1	Espen	84,5	31.01.2022	7		Create a test project to check if the functionality is done based on the research
	A14	Complete feature list of old system, planning of porting	A1	Mari	88	10.02.2022	2		Make a complete overview of the functionality and plan how it can be iteratively implemented in the new system
1	A15	Port old system	A1	Ole-Andre	220,5	11.02.2022	14		system. Add one feature then test and verify before adding the next.
1	A16	Verify that FAT works	A1	Espen	6,5	24.02.2022	2		Complete a Factory acceptance test, using the new system. Check if the functionality is the same
2	A2	Multiple thruster models		Mari	394,5	26.02.2022	28		Add two new thruster models to the simulator
2	A21	Research thrusters	A2	Ole-Andre	28,5	26.02.2022	7		Research how the thrusters work, how they can be modelled, any simplifications that can be done
2	A22	Sketch thruster models, plan the making	A2	Espen	18,5	03.03.2022	2		Make a plan for the development process and divide and clarify the requirements.
2	A23	Make model thruster typ1	A2	Mari	153	05.03.2022	14		
2	A24	Test thruster typ1	A2	Ole-Andre	26,5	19.03.2022	5		Test and verify that the model works as expected, test with IO
2	A25	Make model thruster typ2	A2	Espen	17	14.03.2022	14		
2	A26	Test thruster typ2	A2	Mari	44,5	28.03.2022	5		Test and verify that the model works as expected, test with IO
3	A3	Improve models		Ole-Andre	0	02.04.2022	28		Improve the models used in the simulator
3	A31	AC motor model	A3	Espen	0	02.04.2022	14		Make new and improved model for a AC motor
3	A32	PM motor model	A3	Mari	0	02.04.2022	14		Make new and improved model for a PM motor
3	A33	FPP models	A3	Ole-Andre	0	16.04.2022	14		Make new and improved model for FP propeller
3	A34	CPP models	A3	Espen	0	16.04.2022	14		Make new and improved model for CP propeller
5	A5	Thesis report		Espen	512,5	24.01.2022	117		
5	A51	Draft A1	A5	Mari	30,5	19.02.2022	7		Add the info from phase 1 to the report
5	A52	Draft A2	A5	Ole-Andre	213	19.03.2022	14		Add the info from phase 2 to the report
5	A53	Draft A3	A5	Espen	260	10.04.2022	20		Add the info from phase 3 to the report
5	A54	Complete Report	A5	Mari	0	06.05.2022	14		Finalize the report
2	A28	OPC communication and logger	A2	Ole-Andre	61,5	28.03.2022	5		Make a logger in python that can be used to logg and document the results, and enable the OPC communication with the system
2	A29	Azimuthing function	A2	Espen	20,5	29.03.2022	6		Make and integrate the azimuthing function in Matlab, Simulink, 4diac and CAN on ESP
5	A56	Make presentation	A2	Espen	21,5	29.03.2022	6		Make the movie fo the presentation and plan the presentation
5	A57	Make poster	A2	Mari, Ole-Andre	3	29.03.2022	6		Plan and make the poster

Hour list

Task	Date	Name	Hours	Comment
A11	25.01.2022	Espen	4	Installing 4diac IDE for Mac, there is a java issue making it unable to open
A11	26.01.2022	Espen	1,5	Installing 4diac IDE for Windows
A11	27.01.2022	Espen	4	Local simulation and general overview of the application
A11	27.01.2022	Ole	5	Research and simulation
A02	27.01.2022	Ole	3,5	Meeting with steering group and discussing future progression
A11	27.01.2022	Mari	4	Research 4diac and test simulation
A02	27.01.2022	Mari	1,5	Meeting with steering group and discussion with project group
A03	27.01.2022	Mari	2,5	Prepare project pitch
A02	27.01.2022	Espen	1,5	Meeting with steering group and discussion with project group
A03	27.01.2022	Espen	2	Prepare project pitch
A01	27.01.2022	Espen	0,5	Write meeting minutes for steering group meeting
A03	28.01.2022	Mari	1	Prepare project pitch
A11	28.01.2022	Mari	2	Research and simulation
A03	28.01.2022	Espen	1	Prepare project pitch
A11	28.01.2022	Espen	2	Research and simulation
A03	28.01.2022	Espen	3	Present pitch and listen to other pitches
A03	28.01.2022	Ole	3	Present project pitch
A03	28.01.2022	Ole	2	Prepare project pitch
A03	28.01.2022	Mari	3	Present pitch and listen to other pitches
A02	28.01.2022	Mari	1	Internal meeting inc. progress report
A02	28.01.2022	Espen	1	Internal meeting inc. progress report
A02	28.01.2022	Ole	1	Internal meeting inc. progress report
A1	03.02.2022	Ole	8	Researching and preparing for building runtime environment on WAGO PLC
A11	03.02.2022	Mari	5,5	Research 4diac and functionability
A13	03.02.2022	Mari	3	Check test setup and install Automation Studio
A13	03.02.2022	Espen	4	Setup and testing of versioncontrol for 4diac, we will use github
A11	03.02.2022	Espen	3,5	Overview of any missing reasearch
A12	04.02.2022	Mari	2,5	Research communication 4diac-Matlab
A1	04.02.2022	Ole	7	Troubleshooting setting up runtime environment on Ubuntu
A13	04.02.2022	Mari	4	Testing setup of communcation 4diac-matlab
A13	04.02.2022	Espen	6	Setup of building of runtime env. in Ubuntu, helping matlab communication
A13	05.02.2022	Espen	3	Setup of building of runtime env. in Ubuntu
A13	09.02.2022	Mari	1	Testing setup of communcation 4diac-matlab
A02	10.02.2022	Mari	0,5	Meeting with steering group
A01	10.02.2022	Mari	0,5	Write meeting minutes for steering group meeting
A13	10.02.2022	Mari	6,5	Testing communcation 4diac-matlab, Automation Studio, connect to PLC
A13	10.02.2022	Espen	8	Installing Eclipse, testing connection to the wago plc
A02	10.02.2022	Espen	0,5	Meeting with steering group
A13	10.02.2022	Ole	8	Setup of building of runtime env. in Ubuntu
A02	10.02.2022	Ole	0,5	Installing Eclipse, testing connection to the wago plc
A14	11.02.2022	Mari	4,5	Researching available PLC code snippets from last years project, draft feature list
A02	11.02.2022	Mari	1	Internal meeting inc. progress report
A13	11.02.2022	Espen	6,5	Building and testing forte on wago, successfully used io modules
A02	11.02.2022	Espen	1	Internal meeting inc. progress report
A13	11.02.2022	Ole	8	Troubleshooting and bulding runtime environment
A02	11.02.2022	Ole	0,5	Internal meeting inc. progress report
A14	12.02.2022	Mari	1,5	Plan structure of imported project in 4diac
A01	12.02.2022	Espen	0,5	Update gantchart and add to progress report
A13	17.02.2022	Ole	6	Testing and mapping functionalities such as analog and digital I/O
A51	17.02.2022	Ole	3	Started importing packages and drafting
A13	17.02.2022	Espen	9	Testing Analog IO for wago plc running forte, creating FB for testing of self made FBs in Forte
A13	18.02.2022	Espen	9,5	Testing Analog IO for wago plc running forte, creating stup in Ubuntu for building forte for Linux for testing of FBs. Testing external FBs on Wago with success.
A51	18.02.2022	Ole	8	Creating a basic structure and began to lightly fill in chapters, while being on standby assistance
A14	18.02.2022	Mari	0,5	Looking at Java code
A13	19.02.2022	Espen	1	Create an overview of how to create and test FBs on WAGO
A13	22.02.2022	Espen	1	Testing and building of Java code for FAT
A14	24.02.2022	Mari	7	Creating feature list for BR code and understanding the system
A02	24.02.2022	Mari	0,5	Meeting with steering group
A14	24.02.2022	Espen	7	Creating feature list for BR code and understanding the system
A02	24.02.2022	Espen	0,5	Meeting with steering group
A14	24.02.2022	Ole	7	Creating feature list for BR code and understanding the system
A14	25.02.2022	Espen	7	Planning of the structure of the program, and testing B&R program
A14	25.02.2022	Mari	6,5	Planning of the structure of the program
A02	25.02.2022	Mari	1	Internal meeting inc. progress report
A02	25.02.2022	Espen	1	Internal meeting inc. progress report
A02	25.02.2022	Ole	1	Internal meeting inc. progress report
A14	25.02.2022	Ole	7	Planning of the structure of the program

A02	25.02.2022	Ole	0,5	Write meeting minutes for steering group meeting
A14	03.03.2022	Espen	8	Continue with the planning of the porting, there is still a problem with the java code (this is a low priority at the moment)
A14	03.03.2022	Mari	8	Continue planning of porting
A51	03.03.2022	Ole	8	Thesis writing
A14	04.03.2022	Espen	4	When trying to start the porting process it was noticed that we did not completly understand the functionality and had to be redone, setup of github and make a simple name convention for coding
A15	04.03.2022	Espen	4	Start the porting, make first version of the thruster sim block
A14	04.03.2022	Mari	4	Replan porting, testing push/pull to github from 4diac, coding convention
A15	04.03.2022	Mari	4	Start the porting, make first version of pump/drive control. Problem with internal FBs when compiling to Forte.
A02	10.03.2022	Mari	1	Meeting with steering group
A03	10.03.2022	Mari	0,5	Write meeting minutes for steering group meeting
A02	10.03.2022	Espen	1	Meeting with steering group
A03	10.03.2022	Espen	0,5	Write meeting minutes for steering group meeting
A15	10.03.2022	Espen	6	Make function block for thruster sim, and testing of integration of matlab code in 4diac
A15	10.03.2022	Mari	6	Porting of pump and drive control to 4diac
A15	11.03.2022	Espen	6,5	Make function block for remapping, and testing of function blocks
A02	11.03.2022	Espen	1	Internal meeting inc. progress report
A15	11.03.2022	Mari	6,5	Porting of JS and DP to 4diac
A02	11.03.2022	Mari	1	Internal meeting inc. progress report
A15	11.03.2022	Ole	6,5	Catching up after sickness and porting of visualisation
A02	11.03.2022	Ole	1	Internal meeting inc. progress report
A15	17.03.2022	Mari	4	Creating signal list to Wago PLC, defining needed equipment
A15	17.03.2022	Mari	2	Compiling Forte for all FBs currently on main branch, FBTesting
A03	17.03.2022	Mari	1,5	Logging and documentation
A15	17.03.2022	Espen	2	Bug fix for remap FB, testing and verify that it works as intended
A15	18.03.2022	Mari	7,5	Reverse remap FB, start main program in 4diac, version control issues
A16	18.03.2022	Espen	3	Testing the Java FAT on the B&R controller, it is not working and the program cant connect to the controller (the connection works in Python and OPC Viewer)
A15	18.03.2022	Espen	5	Updating docs for porting, testing OPC UA in 4diac. Has to compile the library seperately! Fixing som problems in git...
A15	18.03.2022	Ole	8	Testing of function blocks and created termination list
A01	23.03.2022	Espen	3,5	Enable progress bars in the timeline, and sending out agenda for the meeting
A02	24.03.2022	Mari	0,5	Meeting with steering group
A03	24.03.2022	Mari	0,5	Write meeting minutes for steering group meeting
A15	24.03.2022	Mari	7	Fix bugs on main program, IO addressing list, IO addressing in 4diac
A02	24.03.2022	Ole	0,5	Meeting with steering group
A15	24.03.2022	Ole	7,5	Verified connections and began wiring up the new wago system
A16	24.03.2022	Espen	3,5	Continue with testing of the FAT-program, and tried to debug it
A02	24.03.2022	Espen	0,5	Meeting with steering group
A15	24.03.2022	Espen	4	Automate the making of Forte and sending it to wago PLC, adding adresses to the cable list
A15	25.03.2022	Espen	7,5	Automate the making of local Forte, started testing of the main program
A02	25.03.2022	Espen	1	Internal meeting inc. progress report
A02	25.03.2022	Mari	1	Internal meeting inc. progress report
A15	25.03.2022	Mari	5,5	Correcting main program, testing main program on hardware
A15	25.03.2022	Ole	7,5	Assissting in function block testing and troubleshooting
A02	25.03.2022	Ole	1	Internal meeting inc. progress report
A51	26.03.2022	Mari	1	Thesis writing
A15	28.03.2022	Mari	5,5	Write documentation for FBs, assist in testing
A21	28.03.2022	Mari	2	Research both thruster types
A15	28.03.2022	Ole	7,5	Looking into running scripts on the cpu and assisting in testing and troubleshooting
A51	28.03.2022	Ole	1	Thesis writing
A51	28.03.2022	Espen	0,5	Thesis writing
A15	28.03.2022	Espen	8	Faultfinding on use of anaolog out with external DO, and trying to autostart forte on wago
A15	29.03.2022	Mari	4,5	Testing and debugging 4diag/wago
A21	29.03.2022	Mari	3	Research both thruster types
A15	29.03.2022	Ole	5,5	Assisting in testing and debugging in 4diac/wago but also conversing with wago support regarding the boot issues
A51	29.03.2022	Ole	1	Thesis writing
A51	29.03.2022	Mari	0,5	Thesis writing
A15	30.03.2022	Ole	2	Establishing and finishing automatic forte boot
A21	30.03.2022	Mari	4,5	Research both thruster types
A15	30.03.2022	Mari	3,5	Assisting in testing and debugging in 4diac/wago
A15	30.03.2022	Ole	2	Assisting in testing and debugging in 4diac/wago
A21	30.03.2022	Ole	4	Researching tunnel thrusters
A51	30.03.2022	Mari	0,5	Thesis writing
A51	30.03.2022	Ole	1,5	Thesis writing
A15	30.03.2022	Espen	7,5	Faultfinding on the control system. The AO for RPM feedback was connected the wrong way. A few alarm signals was not connected and there was a few address that was wrong in the program

A51	30.03.2022	Espen	0,5	Thesis writing
A15	31.03.2022	Ole	6	Troubleshooting, bugfixing of program and redoing some wiring.
A21	31.03.2022	Ole	2	Research tunnel thruster
A51	31.03.2022	Ole	1	Thesis writing
A21	31.03.2022	Mari	3,5	Research both thruster types
A22	31.03.2022	Mari	1,5	Planning adding of thruster types
A15	31.03.2022	Mari	3	Bug fix for rmp feedback in js and dp, assisting in testing
A15	31.03.2022	Espen	4	Buliding and testing new version with bugfixes for thruster sim.
A1	31.03.2022	Espen	2	Testing setup of new 4diac with nightly build
A1	01.04.2022	Espen	4	Testing setup of new 4diac with nightly build
A15	01.04.2022	Espen	6	Port old thruster dynamics to 4diac, and make project in matlab using git
A02	01.04.2022	Espen	1	Internal meeting inc. progress report
A02	01.04.2022	Mari	1	Internal meeting inc. progress report
A02	01.04.2022	Ole	1	Internal meeting inc. progress report
A22	01.04.2022	Mari	4	Planning adding of thruster types
A21	01.04.2022	Mari	2	Research both thruster types
A15	01.04.2022	Ole	2	Rewiring and installing of terminal blocks
A21	01.04.2022	Ole	5	Thruster and modeling research
A1	02.04.2022	Espen	2	Testing setup of new 4diac with nightly build, there is multiple errors when trying to start the build
A15	04.04.2022	Ole	7,5	Began creating a python OPC-client
A51	04.04.2022	Ole	1	Thesis writing
A21	04.04.2022	Mari	2,5	Research dynamic thruster models and cpp model
A22	04.04.2022	Mari	5	Planning adding of thruster types
A52	04.04.2022	Mari	1	Thesis writing
A1	04.04.2022	Espen	3	Testing setup of new 4diac with nightly build, it is decided that it is not worth the effort to spend more time on this
A14	04.04.2022	Espen	5	Building Forte for wago to enable OPC UA, it is unable to build the program
A22	05.04.2022	Mari	4,5	Planning adding of thruster types
A23	05.04.2022	Mari	3	Starting thruster type 1, FB for swing-up
A14	05.04.2022	Espen	6	Building Forte for wago to enable OPC UA,trying to enable the library in menuconfig but ran out of memory the first time and when this was fixed I had a compile error on forte.
A15	05.04.2022	Espen	4	Faultfinding on the FAT-java program and trying to determent why it is not working as expected with the old system
A15	05.04.2022	Ole	7,5	Experimenting with OPC UA through python
A51	05.04.2022	Espen	1	Thesis writing
A14	06.04.2022	Espen	5	Building Forte for wago to enable OPC UA, after fixing the compile error and the image was built again there was no change in the OPC functionality and still recive file format not recognized when enabling OPC in cmake
A15	06.04.2022	Espen	1	Faultfinding on the FAT-java program and trying to determent why it is not working as expected with the old system
A15	06.04.2022	Ole	7,5	Experimenting with OPC UA through python
A51	06.04.2022	Ole	1	Thesis writing
A51	06.04.2022	Espen	1	Thesis writing
A23	06.04.2022	Mari	7,5	FB for swing-up, CPP model in Matlab
A23	07.04.2022	Mari	7	CPP model in Matlab
A02	07.04.2022	Mari	0,5	Meeting with steering group
A52	07.04.2022	Mari	0,5	Thesis writing
A15	07.04.2022	Ole	6,5	OPC Python client
A15	07.04.2022	Espen	5	Trying to enable OPC UA on wago PLC
A02	07.04.2022	Espen	0,5	Meeting with steering group
A22	07.04.2022	Espen	3	Plan and find parts to enable CAN-communication for the simulator
A02	07.04.2022	Ole	0,5	Meeting with steering group
A01	07.04.2022	Ole	0,5	Writing meeting minutes
A23	08.04.2022	Mari	6,5	CPP model in Matlab, research dynamics
A02	08.04.2022	Mari	1	Internal meeting inc. progress report
A02	08.04.2022	Espen	1	Internal meeting inc. progress report
A02	08.04.2022	Ole	1	Internal meeting inc. progress report
A15	08.04.2022	Ole	7,5	OPC Python client
A27	08.04.2022	Espen	6	Trying to enable OPC UA on wago PLC
A23	08.04.2022	Espen	0,5	Faultfinding in Simulink for the CPP simulation
A27	09.04.2022	Espen	5	Install forte on Rapsberry pi and test with OPC, started evaluating docker alternative
A52	10.04.2022	Ole	1	Thesis writing
A52	10.04.2022	Mari	1	Thesis writing
A52	10.04.2022	Espen	2	Thesis writing
A23	11.04.2022	Mari	7,5	CPP model in Matlab, research dynamics
A52	11.04.2022	Mari	1	Thesis writing
A28	11.04.2022	Ole	7,5	Finished OPC UA prototype
A28	11.04.2022	Espen	4	Testing running forte compiled on R-Pi on Wago, testing OPC on rasp-pi
A23	11.04.2022	Espen	3,5	Finding a good thruster model that is cpp compatible
A52	11.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A23	12.04.2022	Mari	7,5	CPP model in Matlab, research dynamics
A52	12.04.2022	Ole	6,5	Thesis writing whilst assisting other members
A23	13.04.2022	Espen	7,5	Testing and researching cpp model for thruster

A23	13.04.2022	Mari	6	CPP model in Matlab, research dynamics
A52	13.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A28	14.04.2022	Espen	4	Testing docker on rasp-pi with custom setup
A28	15.04.2022	Espen	4	Testing docker on rasp-pi with custom setup
A23	16.04.2022	Espen	4	Get a overview of the dynamics of how the new model is
A28	17.04.2022	Espen	4	Added OPC objects for multiple object according to the list
A52	18.04.2022	Espen	3,5	Made table for all the testing of IO modules
A23	19.04.2022	Espen	4	Find MSMC of the thruster model, find info about the thrusters, make simplified propeller inertia calculations
A28	19.04.2022	Espen	4	Added OPC linking and communication to the proram, should be ready for testing
A52	19.04.2022	Espen	1	Write about the adressing of modules, and make a draft of the structure for the method chapter
A23	19.04.2022	Mari	7	Working on new model in simulink, finding numbers for specific thruster
A52	19.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A23	20.04.2022	Mari	7,5	Working on new model in simulink, fixing, improving
A23	20.04.2022	Espen	4,5	Create and modify propeller modell i simulink
A28	20.04.2022	Espen	4	Testing docker on wago with OPC
A52	20.04.2022	Espen	0,5	Think through the general setup of the repport and what it should include
A52	20.04.2022	Ole	6,5	Thesis writing whilst assisting other members
A02	21.04.2022	Mari	1	Meeting with steering group
A23	21.04.2022	Mari	6,5	Improving model in Simulink, adding and tuning pid, adding pitch
A52	21.04.2022	Ole	6,5	Thesis writing whilst assisting other members
A02	21.04.2022	Ole	1	Meeting with steering group
A23	21.04.2022	Espen	3,5	Modify and improve propeller modell i simulink
A28	21.04.2022	Espen	3	Testing OPC on wago/raspi there are some changes that need to be made before it is complete
A01	21.04.2022	Espen	1	Writing meeting minutes
A02	21.04.2022	Espen	1	Meeting with steering group
A52	22.04.2022	Ole	3,5	Thesis writing whilst assisting other members
A23	22.04.2022	Mari	3,5	Modify, improve, and tune propeller model in Simulink
A5	22.04.2022	Mari	3	Reorganize thesis report structure
A23	22.04.2022	Espen	3,5	Modify and improve propeller modell i simulink
A5	22.04.2022	Espen	3	Reorganize thesis report structure
A02	22.04.2022	Mari	1	Internal meeting inc. progress report
A02	22.04.2022	Espen	1	Internal meeting inc. progress report
A02	22.04.2022	Ole	1	Internal meeting inc. progress report
A5	22.04.2022	Ole	3	Reorganize thesis report structure
A28	22.04.2022	Espen	2	Setup Docker with outgoing communication and external volume for boot-files
A23	23.04.2022	Mari	1,5	Simulink model, final pid-tuning, preparing for discretisizing
A28	23.04.2022	Espen	8	Setup Docker with outgoing communication and external volume for boot-files
A28	24.04.2022	Espen	3	Setup Docker with outgoing communication and external volume for boot-files
A1	24.04.2022	Espen	3	Testing and extending the bash program to semi-automate the build process
A23	25.04.2022	Mari	7	Discretize, export to PLC code, import to 4diac, try to make it work
A52	25.04.2022	Espen	2	Thesis writing about the method used for the setup and workflow
A52	25.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A23	26.04.2022	Mari	7,5	Error fixing on discrete cpp model, porting to 4diac and testing
A29	26.04.2022	Mari	0,5	Discuss plans and progress for CANopen solution
A52	26.04.2022	Ole	7	Thesis writing whilst assisting other members
A29	26.04.2022	Ole	0,5	Discuss plans and progress for CANopen solution
A29	26.04.2022	Ole	1	Researching CANopen possibilities
A28	26.04.2022	Espen	5,5	Testing OPC on wago/raspi. Testing of docker container with communication
A29	26.04.2022	Espen	2	Find info about Can and canopen implementation. Discuss plans and progress for CANopen solution
A52	27.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A29	27.04.2022	Espen	1	Discuss plans and progress for CANopen solution, trying to contact Ottar and Robin about the plan
A28	27.04.2022	Espen	4	Test OPC logger with forte
A23	27.04.2022	Espen	5	Testing different discretation methods and check why there is a bigger discrepancy than expected
A23	27.04.2022	Mari	8,5	Making matlab model discrete, retuning, exporting to 4diac, error fixing in 4diac
A52	27.04.2022	Ole	7,5	Thesis writing whilst assisting other members
A23	28.04.2022	Mari	4	Making FBs in 4diac for new CPP dynamic, pitch, and wrapper
A22	28.04.2022	Mari	0,5	Adjust plan for making of thruster models
A25	28.04.2022	Mari	4	Starting system and main apps for azipull thruster in 4diac
A01	28.04.2022	Espen	0,5	Planning and finding the best way to devide the work needed to be finished with A2
A29	28.04.2022	Espen	6	Made the mathematical model of the azimuth function and implemented it as a continuous model in simulink/matlab
A28	28.04.2022	Espen	4,5	Testing writing opc values to the PLC, the initial OPC implementation is now finished
A23	28.04.2022	Ole	1	Added additional I/O for functions
A52	28.04.2022	Ole	5,5	Thesis writing whilst assisting other members
A26	29.04.2022	Ole	6,5	Review documentation and prepared for rewire
A29	29.04.2022	Espen	6	Making the discrete model and implementing it in 4diac
A24	29.04.2022	Espen	3	Assisting in finding the testing methodology
A02	29.04.2022	Espen	1	Internal meeting inc. progress report
A02	29.04.2022	Ole	1	Internal meeting inc. progress report

A52	30.04.2022	Espen	6	Write about the building of forte on wago
A52	01.05.2022	Espen	7	Write about the building of forte on wago, and theory about the DC motor and propeller dymamics
A1	02.05.2022	Espen	5	Remake the thruster model in matlab and in 4diac
A23	02.05.2022	Espen	3,5	Make the circuit and TF for RC filter with voltage divider
A52	02.05.2022	Espen	1,5	Write about the RC circuit
A52	02.05.2022	Ole	7,5	Preparing and documenting expantion of the physical setup
A03	03.05.2022	Espen	5	Div changes in github files and documentation
A24	03.05.2022	Espen	3	Helping with test methodology
A52	03.05.2022	Espen	1,5	Write about the RC circuit
A25	03.05.2022	Mari	7,5	Making main apps for azipull thruster in 4diac
A52	03.05.2022	Ole	7,5	Expantion of physical setup
A29	04.05.2022	Espen	3,5	Make the test circuit for the system (from PWM to 0-10V)
A26	04.05.2022	Espen	4	Develop the test circuit used to make the PWM signal. Faultfing on the wiring, and general test setup.
A52	04.05.2022	Espen	0,5	Thesis writing about the wiring, and commenting
A23	04.05.2022	Espen	3	Adding OPC objects to the new thruster model
A25	04.05.2022	Mari	2,5	Finishing azipull app in 4diac, adding IO addresses
A26	04.05.2022	Mari	5	Wiring new IO, wiring test jig, making test setup, documenting IO addresses, troubleshoot test jig
A52	04.05.2022	Mari	0,5	Thesis writing
A52	04.05.2022	Ole	7,5	Preparing for testing incl troubleshooting
A02	05.05.2022	Mari	0,5	Meeting with steering group
A01	05.05.2022	Mari	1	Writing meeting minutes
A52	05.05.2022	Mari	2,5	Thesis writing, general, thruster model
A26	05.05.2022	Mari	4,5	Testing and troubleshooting, assisting
A02	05.05.2022	Ole	0,5	Meeting with steering group
A26	05.05.2022	Ole	7	Testing, setting up and troubleshooting
A02	05.05.2022	Espen	0,5	Meeting with steering group
A26	05.05.2022	Espen	3,5	Testing of the main program for the azipull thruster
A23	05.05.2022	Espen	7	Changed the thruster dynamics
A52	06.05.2022	Mari	4	Thesis writing about thruster model
A26	06.05.2022	Mari	1,5	Testing azipull system on physical equipment
A25	06.05.2022	Espen	3	Changed the thruster dynamics in 4diac
A26	06.05.2022	Espen	3,5	Testing of the main program for the azipull thruster
A23	06.05.2022	Mari	2,5	Fixing bugs in swing-up FB and swing-up wrapper FB, making system for swing up in 4diac
A02	06.05.2022	Mari	0,5	Internal meeting inc. progress report
A02	06.05.2022	Espen	0,5	Internal meeting inc. progress report
A02	06.05.2022	Ole	0,5	Internal meeting inc. progress report
A26	06.05.2022	Ole	5	Testing of the main program for the azipull thruster
A52	06.05.2022	Ole	3	Thesis writing
A26	06.05.2022	Espen	4	Remake the pitch disc model
A23	07.05.2022	Mari	2	Making system for swing up in 4diac
A52	07.05.2022	Espen	8	Thesis writing
A52	08.05.2022	Mari	4,5	Thesis writing
A52	08.05.2022	Espen	8	Thesis writing
A24	09.05.2022	Espen	12	Gathering data and making graphs
A24	09.05.2022	Mari	1,5	Testing swing-up on physical equipment
A52	09.05.2022	Mari	6,5	Thesis writing
A52	09.05.2022	Ole	7,5	Thesis writing while assisting where needed
A52	10.05.2022	Mari	8,5	Thesis writing
A52	10.05.2022	Espen	7	Thesis writing in results
A24	10.05.2022	Espen	7	Gathering data from simulations and make graphs
A52	10.05.2022	Ole	7,5	Thesis writing
A53	10.05.2022	Espen	14	Thesis writing
A52	11.05.2022	Ole	8	Thesis writing
A53	11.05.2022	Mari	8,5	Thesis writing
A53	11.05.2022	Espen	14	Thesis writing
A52	12.05.2022	Ole	4,5	Thesis writing
A02	12.05.2022	Ole	5,5	Meeting with steering group and additonal planning regarding the thesis
A02	12.05.2022	Mari	5,5	Meeting with the steering group and internal meeting regarding report
A53	12.05.2022	Mari	4,5	Thesis writing
A53	12.05.2022	Espen	10	Thesis writing
A53	13.05.2022	Mari	10,5	Thesis writing, thesis walkthrough
A53	13.05.2022	Ole	11	Thesis writing
A53	13.05.2022	Espen	12	Thesis writing
A57	14.05.2022	Mari	3	Make poster
A53	14.05.2022	Mari	3	Thesis writing
A53	14.05.2022	Espen	13	Thesis writing
A53	14.05.2022	Ole	5	Thesis writing
A53	15.05.2022	Espen	13	Thesis writing
A53	15.05.2022	Mari	7	Thesis writing, walkthrough and discussion
A53	15.05.2022	Ole	8	Thesis writing, walkthrough and discussion
A56	16.05.2022	Mari	6,5	Filming for presentation
A56	16.05.2022	Espen	15	Filming for presentation and editing

A53	16.05.2022	Ole	6,5	Filming for presentation
A53	16.05.2022	Ole	2	Thesis writing
A53	17.05.2022	Mari	8,5	Thesis writing, walkthrough and discussion
A53	17.05.2022	Espen	8,5	Thesis writing, walkthrough and discussion
A53	17.05.2022	Ole	8	Thesis writing
A53	18.05.2022	Mari	10	Thesis writing, walkthrough and discussion
A53	18.05.2022	Ole	14	Thesis writing
A53	18.05.2022	Espen	15	Thesis writing, walkthrough and discussion
A53	19.05.2022	Mari	14	Thesis writing, appendices
A53	19.05.2022	Ole	14	Thesis writing
A53	19.05.2022	Espen	14	Thesis writing, appendices
A53	20.05.2022	Mari	4	Thesis writing, appendices
A53	20.05.2022	Ole	4	Thesis writing
A53	20.05.2022	Espen	4	Thesis writing, appendices

C Progress reports

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 1
Progress report	Period/week(s) 1/1	Number of hours this period. (from log) Approx. 50	Prosjektgruppe (navn) Gruppe 2	Dato 28.01.2022

Main goal/purpose for this periods work - Research 4diac	
Planned activities this period - Research 4diac - Research 4diac communication	
Actually conducted activites this period - Research 4diac Everyone tried to install and do the tutorial and testing how to do local simulation. There was some testing to get it to work on MacOS (not working yet) Research how 4diac works and the structure of programs and features - Research 4diac communication - Make pitch	
Description of/ justification for potential deviation between planned and real activities - Due to extra activities such as "Endelig fredag"'s pitch presentation the group chose to focus on this - Establishing groups and mandatory assignments from the subject "Ingeniørfaglig Systemtenkning"	
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report - Due to the low number of working hours each week it is decided to only make a progress report bi-weekly instead of weekly	
Main experience from this period - It is possible to use MQTT and OPC UA with 4diac. - There exist possibilities of making a GUI using 4diac - Eclipse NeoSCADA is an option for GUI - Learn the structure and implementation of FBs and dividing resources in 4diac	
Main purpose/focus next period - Test setup	
Planned activities next period - Continue research of 4diac - Continue research of 4diac communication - Test setup (test program on wago, setup rack) - Complete feature list of old system, planning of porting	
Other	
Wish/need for counceling - Nothing in particular	
Approval/signature group leader Ole-Andre Aurvåglid Ramstad	Signature other group participants Espen Brandtzæg Godø Mari Ellinor Longva Haram

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 2
Progress report	Period/week(s) 2/2	Number of hours this period. (from log) Approx. 96	Prosjektgruppe (navn) Gruppe 2	Dato 11.02.2022

Main goal/purpose for this periods work
- Test setup (test program on Wago, setup rack) - Complete feature list of old system, planning of porting
Planned activities this period
- Continue research of 4diac - Continue research of 4diac communication - Install Eclipse Forte on Wago PLC - Test communication from 4diac to other platforms - Plan porting of system from B&R to 4diac
Actually conducted activites this period
- Researched Forte for Wago PLC - Researched 4diac communication with Matlab - Received and examined test rack - Tested communication between 4diac and Matlab using an open source library. Tested with server in Matlab. - Installed Forte to Wago PLC and tested communication to PC It is not tested with custom FBs. - Tested IO from Wago PLC to 4diac Digital IO works as expected, and Analog IO is not tested yet - Draft of feature list for porting of system from B&R to 4diac
Description of/ justification for potential deviation between planned and real activities
- Due to problems when compiling Forte for Wago PLC there were multiple issues we had to solve, without much documentation. Further this had to be done in Ubuntu. Thus, this took an extensive amount of time. - Without the source code from last year's project, the feature list and planning of porting is difficult to complete. Arrangements for accessing the source code are being made.
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report
- The period for testing setup will be expanded by 14 additional days (until 18.02) due to delays during testing. - Start of the period for porting of the system is moved from 14.02 to 18.02 due to delays during testing and delivery of source code, see Figure 1.
Main experience from this period
- 4diac can communicate with Matlab using the open source library, but the Matlab version can be no newer than R2020a - Fault finding on building and distributing with ptxdist, and eclipse CDT. - Following outdated guides that require changes based on system specific problems and solutions. - Discovered the differences between IEC 61131-3 (which the current PLC project is using) and IEC 61499 (which 4diac is using).
Main purpose/focus next period
- Complete test setup - Plan porting of system - Start porting of system
Planned activities next period

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 2 av 2
Progress report	Period/week(s) 2/2	Number of hours this period. (from log) Approx. 96	Prosjektgruppe (navn) Gruppe 2	Dato 11.02.2022

- Continue test setup (test program on Wago with analog IO, custom FBs)
- Continue complete feature list of old system, planning of porting
- Start porting of the system

Other

Wish/need for counceling

- Nothing in particular

Approval/signature group leader	Signature other group participants
Espen Brandtzæg Godø	Mari Ellinor Longva Haram Ole-Andre Aurvåglid Ramstad

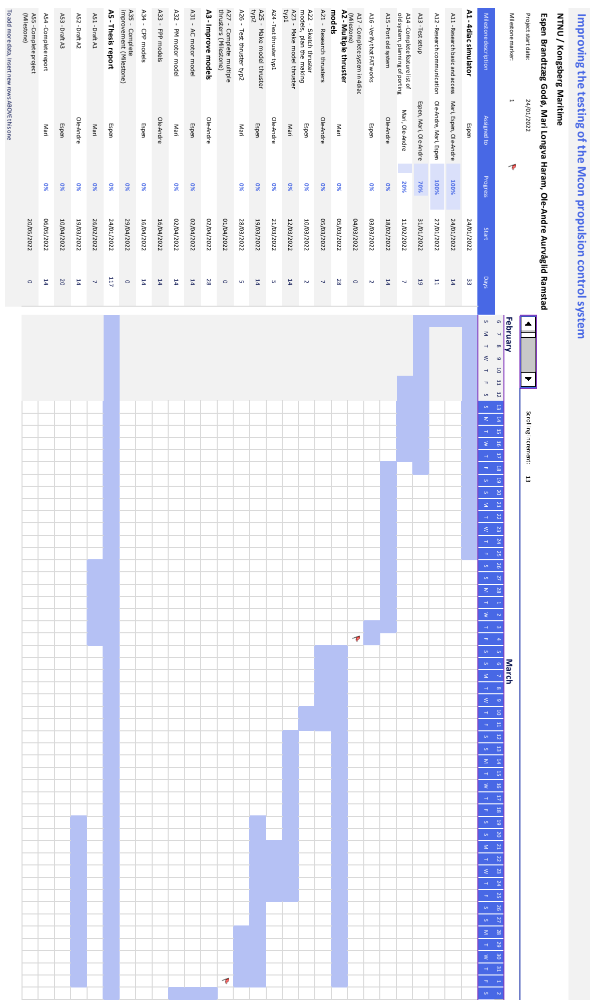


Figure 1: Updated Gnatchart

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 2
Progress report	Period/week(s) 3/2	Number of hours this period. (from log) Approx. 86	Prosjektgruppe (navn) Gruppe 2	Dato 25.02.2022

Main goal/purpose for this periods work
- Plan porting of old system - Start porting of old system
Planned activities this period
- Continue test setup (test program on Wago with analog IO, custom FBs) - Continue complete feature list of old system, planning of porting - Start porting of the system
Actually conducted activites this period
- Completed test setup - Continued feature list of old system - Continued planning of porting the old system
Description of/ justification for potential deviation between planned and real activities
- One of the project group members got sick with Covid-19 which gave an overall small delay - Lack of access to the source code delayed both the planning of the system porting and the porting itself - Underestimated the amount of work needed to plan the porting
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report
- Start of the period for porting of the system is moved from 18.02 to 03.03 due to delays during testing and delivery of source code, see Figure 1. Due to this, the following periods were also shifted.
Main experience from this period
- Addressing of I/O modules in 4Diac. - How to build Forte with custom FBs for Wago and Linux using CMake and ptxdist. - The version of B&R Automation Studio needed to access the previous project is unstable and crashes a lot. - There is a discrepancy between the source code which was provided and the one that is currently on the PLC. - Planning to transfer from one IEC standard to a higher-level standard is proving more challenging when designing the system than previously assumed. - Creating a visual representation of a system makes it easier to both plan and understand the system.
Main purpose/focus next period
- Finish planning the porting - Finish porting - Begin verifying the FAT still works after porting.
Planned activities next period
- Make I/O list - Make/change Matlab program - Make function blocks for thruster, pumps, mode selection and feedback - Replace B&R modules with Wago
Other
Wish/need for counceling

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 2 av 2
Progress report	Period/week(s) 3/2	Number of hours this period. (from log) Approx. 86	Prosjektgruppe (navn) Gruppe 2	Dato 25.02.2022

- Nothing in particular

Approval/signature group leader	Signature other group participants
Mari Ellinor Longva Haram	Ole-Andre Aurvåglid Ramstad Espen Brandtzæg Godø

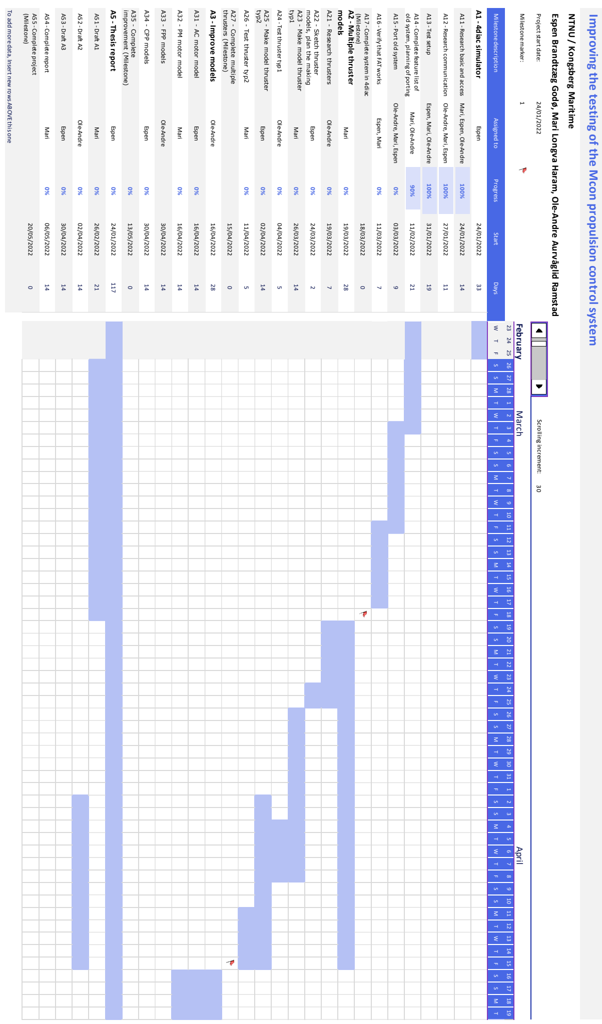


Figure 1: Updated Ganttchart

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 2
Progress report	Period/week(s) 4/2	Number of hours this period. (from log) Approx. 80	Prosjektgruppe (navn) Gruppe 2	Dato 11.03.2022

Main goal/purpose for this periods work
- Finish planning the porting - Finish porting - Begin verifying the FAT still works after porting.
Planned activities this period
- Make I/O list - Make/change Matlab program - Make function blocks for thruster, pumps, mode selection and feedback - Replace B&R modules with Wago
Actually conducted activites this period
- Finished feature list of old system - Finished planning of porting the old system - Started porting of the system - Replanned porting of the old system - Started porting of the old system the new way - Made FBs for thruster, pump, drive, DP, JS and IO previously displayed in the visualisation in Automation Studio.
Description of/ justification for potential deviation between planned and real activities
- One of the project group members got sick with Covid-19 which gave an overall small delay - Misinterpreted the way the program should operate, which caused the need to make a new structure plan for the ported system. This was discovered after the porting had started. - There is a bug in the current version of 4diac/FORTE which makes it impossible to use internal FBs in Structured Text.
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report
- Extended the period for porting of the system to 24.03 due to delays during planning and problems with using internal FBs in 4diac, see Figure 1. Due to this, the upcoming periods in A2 were also shifted.
Main experience from this period
- There will be released a new version of 4diac later in April, and this version should be stable enough in 2 weeks to switch. - How to use composite FBs as an alternative to internal FBs. - How to export models from MATLAB to Structured Text that can be integrated in 4diac.
Main purpose/focus next period
- Finish porting - Verify that the FAT still works after porting. - Research Thrusters
Planned activities next period
- Make I/O list - Finish the function blocks for remap and create system of all FBs - Replace B&R modules with Wago
Other
Wish/need for counceling
- Nothing in particular

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 2 av 2
Progress report	Period/week(s) 4/2	Number of hours this period. (from log) Approx. 80	Prosjektgruppe (navn) Gruppe 2	Dato 11.03.2022

Approval/signature group leader Ole-Andre Aurvåglid Ramstad	Signature other group participants Espen Brandtzæg Godø Mari Ellinor Longva Haram

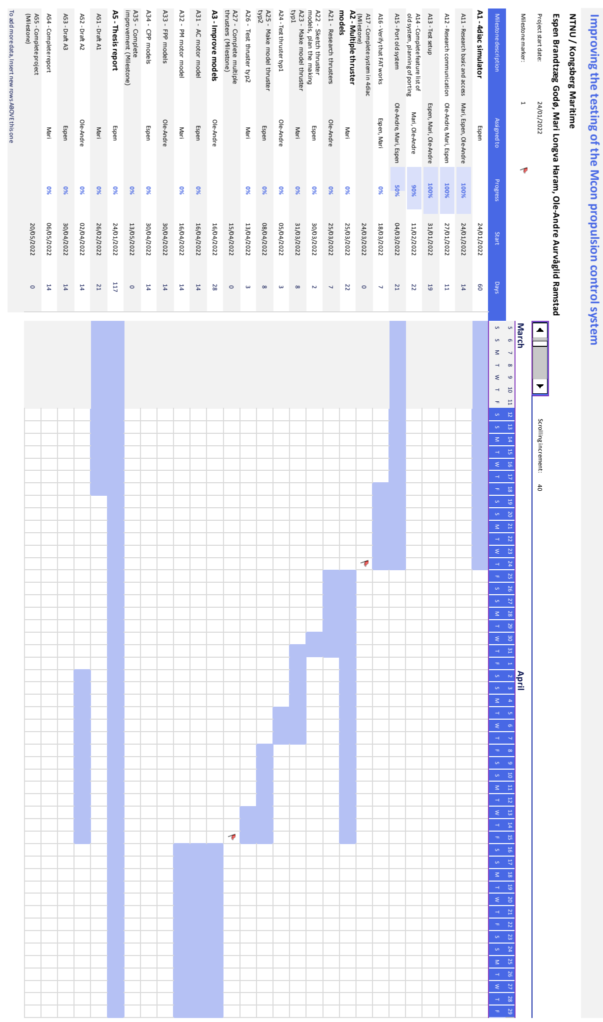


Figure 1: Updated Ganttchart

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 2
Progress report	Period/week(s) 5/2	Number of hours this period. (from log) Approx. 85	Prosjektgruppe (navn) Gruppe 2	Dato 25.03.2022

Main goal/purpose for this periods work

- Finish porting
- Begin verifying the FAT still works after porting.

Planned activities this period

- Make I/O list
- Finish the function blocks for remap and create system of all FBs
- Replace B&R modules with Wago

Actually conducted activites this period

- Finished FBs for remap
- Created system of all FBs
- Began testing FBs on physical components
- Made I/O list
- Replaced B&R modules with Wago
- Updated the Gantt-diagram to include progress in the main part

Description of/ justification for potential deviation between planned and real activities

- There occurred a slight delay due to some members being sick and having to take an exam.

Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report

- Extended the period for porting of the system to 01.04 due to addressing issues within 4diac, see Figure 1. Due to this, the upcoming periods in A2 were also shifted, but the group will start on A2 before A1 is finished to maximise the use of resources.
- The project group has now finished the additional subject INGA2300 and can now dedicate the full work week to the bachelor's project. It is therefore decided that the progress reports will be written at the end of each week instead of every other.

Main experience from this period

- Programming main program and bug finding in 4diac
- Addressing in 4diac is inconsistent with regards to DI and AI
- The Forte runtime environment crashes when there is an error within the program
- How to make bash-scripts to automate the process of building and transferring Forte to Wago PLC

Main purpose/focus next period

- Complete phase one
- Research Thrusters
- Make thruster model for new type

Planned activities next period

- Finish the testing and debugging of the main program
- Sketch thruster models, plan the making
- Make model for thruster type 1
- Test thruster type 1

Other

Wish/need for counceling

- Nothing in particular

Approval/signature group leader	Signature other group participants
Espen Brandtzæg Godø	Mari Ellinor Longva Haram

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 2 av 2
Progress report	Period/week(s) 5/2	Number of hours this period. (from log) Approx. 85	Prosjektgruppe (navn) Gruppe 2	Dato 25.03.2022

	Ole-Andre Aurvåglid Ramstad

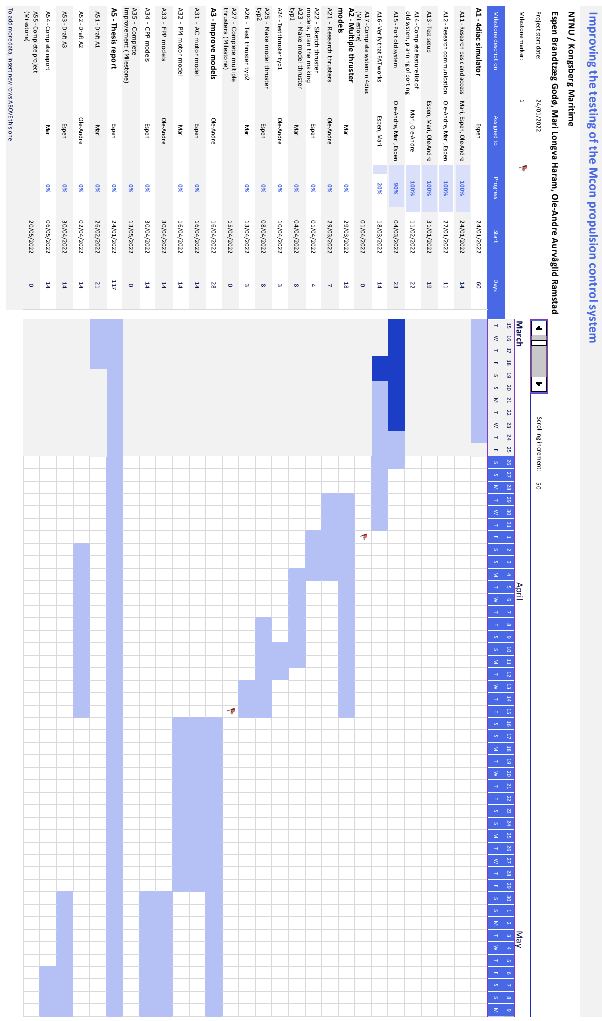


Figure 1: Updated Ganttchart

1) 1 Internal meeting and 1 meeting with Steering group.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 1 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 1 av 3
Progress report	Period/week(s) 6/1	Number of hours this period. (from log) Approx. 115	Prosjektgruppe (navn) Gruppe 2	Dato 01.04.2022

Main goal/purpose for this periods work

- Complete phase one
- Research Thrusters
- Make thruster model for new type

Planned activities this period

- Finish the testing and debugging of the main program
- Sketch thruster models, plan the making
- Make model for thruster type 1
- Test thruster type 1

Actually conducted activities this period

- Finish testing and debugging of the main program
- Created list of all FB’s created for this project in readme file, with description and signal list.
- Completed phase one*
- Researched thrusters
- Started planning the making of the thruster types in 4diac and Matlab
- Identified bugs/problems in the program
- Enabled autoboot to forte runtime

Description of/ justification for potential deviation between planned and real activities

- Lack of documentation on thruster types
- When describing the goals for this period it was not accounted for that the period was only one week

Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report

- (*) It was decided that the FAT java program should be given a lower priority, this should be finished in any unoccupied time that will occur inn phase A2.
- The activity A22 *Sketch thruster models, plan the making* was moved 1 day earlier, since lack of documentation slowed the research activity. Therefore, it was practical to start a plan based on the available documentation. The activity was also extended by two days (until Tuesday 05.04) because of missing thruster documentation from Kongsberg. This is shown in the excerpt of the Gantt chart in figure 1.

Main experience from this period

- How to setup wago PLC to autostart a script on startup
- There is an issue when using more than one 750-513 module at the start of the plc block causing the analog out modules to not work as intended.
- Learned about Kongsberg’s range of azimuth and azipull thrusters.
- How to incorporate git version control in Matlab
- 4Diac IDE’s nightly build is problematic to start but need to be tested more before we can decide whether we should change or wait for the final release.

Main purpose/focus next period

- Plan thruster models
- Make thruster models

Planned activities next period

- Research thrusters
- Sketch thruster models, plan the making

1) 1 Internal meeting.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 1 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 2 av 3
Progress report	Period/week(s) 6/1	Number of hours this period. (from log) Approx. 115	Prosjektgruppe (navn) Gruppe 2	Dato 01.04.2022

- Make dynamic model of swing-up azimuth thruster in Matlab/Simulink - Incorporate dynamic model of swing-up azimuth thruster in 4diac - Start to make model thruster type 2	
Other	
Wish/need for counselling - Nothing in particular	
Approval/signature group leader Mari Ellinor Longva Haram	Signature other group participants Ole-Andre Aurvåglid Ramstad Espen Brandtzæg Godø

1) 1 Internal meeting.

IELEA2920 Bacheloroppgave automatisering	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 1 planned	Firma - Oppdragsgiver NTNU i Ålesund / Kongsberg Maritime	Side 3 av 3
Progress report	Period/week(s) 6/1	Number of hours this period. (from log) Approx. 115	Prosjektgruppe (navn) Gruppe 2	Dato 01.04.2022

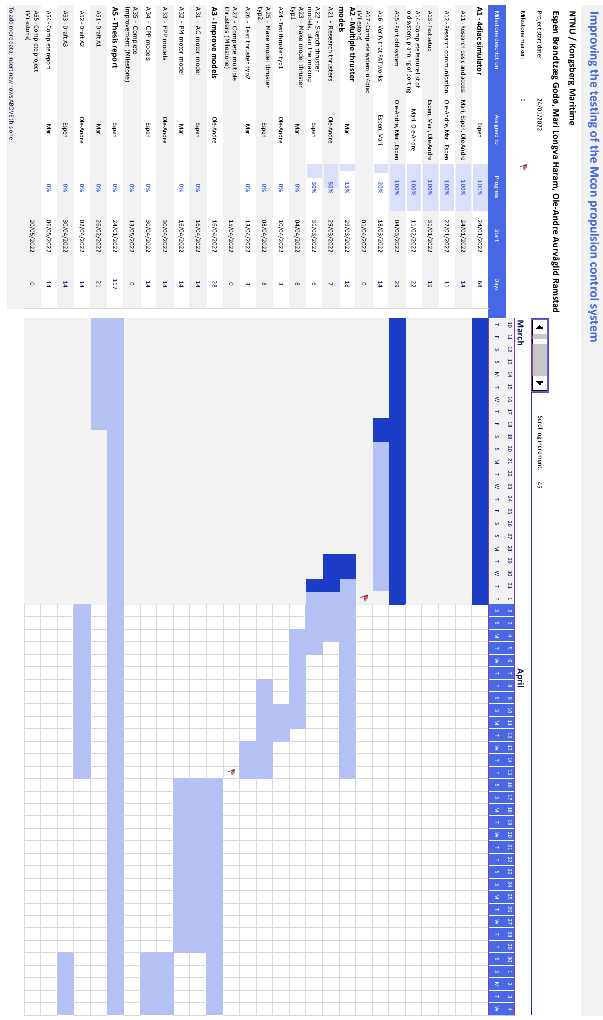


Figure 1: Updated Gantt chart

1) 1 Internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 1 of 3
Progress report	Period/week(s) 7/1	Number of hours this period. (from log) Approx. 125	Project group (name) Group 2	Date 08.04.2022

Main goal/purpose for this periods work

- Plan thruster models
- Make thruster models

Planned activities this period

- Research thrusters
- Sketch thruster models, plan the making
- Make dynamic model of swing-up azimuth thruster in Matlab/Simulink
- Incorporate dynamic model of swing-up azimuth thruster in 4diac
- Start to make model thruster type 2

Actually conducted activities this period

- Continuation of researching thruster types and CPP model for Matlab/Simulink
- Continuation of planning adding of thruster types and CPP model in Matlab/Simulink
- Made FB for Swing-up functionality in 4diac
- Working on the CPP model in Matlab/Simulink
- Began making OPC UA Client in python
- Testing OPC communication in 4diac and forte

Description of/ justification for potential deviation between planned and real activities

- The source the CPP model was planned to be based on was not as good as assumed, this task will therefore need more time to create a model that is useable.
- Communication between a logger and the Wago PLC turned out to be a bigger task than anticipated. The cross compiler could not compile Forte with the OPC UA library and the task of creating a logger with python turned out to be more problematic than expected.

Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report

- The creation of the logger is expanded to be its own task (A28) and includes the time to enable the OPC communication from the system and making of the logger
- A2 had to be reordered, and testing was moved to the end of week 16. A3 was moved and compacted to match the remaining time. The scope of A3 may have to be narrowed because of this. This will be revalued in the next progress report, see Figure 1.
- Because of easter holiday the next progress report will be written at the end of week 16, containing activities from week 15 and 16.

Main experience from this period

- OPC UA’s previous library has been deprecated and been combined with Asyncio. Their combined library lacks documentation, and their examples are hard to understand.
- OPC UA’s hierarchy and node system
- Working with control systems in Simulink and PID tuning.
- Better understanding of using a cross compiler and the setup of ptxdist
- How to enable OPC UA in 4diac Forte

Main purpose/focus next period

- OPC UA communication and logger
- Making and testing new thruster types

Planned activities next period

- Finish python logger
- Finish the OPC setup
- Test and add CAN-bus to the system with an ESP-32

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 2 of 3
Progress report	Period/week(s) 7/1	Number of hours this period. (from log) Approx. 125	Project group (name) Group 2	Date 08.04.2022

- Make model thruster type 1
- Test thruster type 1
- Make model thruster type 2

Other

Wish/need for counselling

- May need counselling for creating the dynamic CPP thruster model in Matlab/Simulink.

Approval/signature group leader	Signature other group participants
Mari Ellinor Longva Haram	Ole-Andre Aurvåglid Ramstad Espen Brandtzæg Godø

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 3 of 3
Progress report	Period/week(s) 7/1	Number of hours this period. (from log) Approx. 125	Project group (name) Group 2	Date 08.04.2022

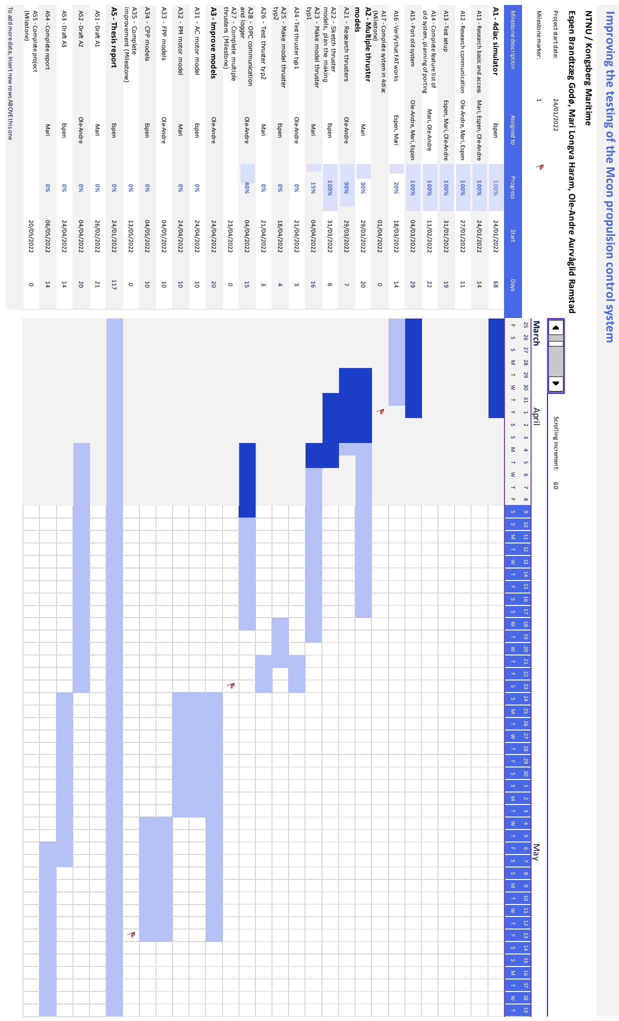


Figure 1: Updated Gantt chart

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project: Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer: NTNU i Ålesund / Kongsberg Maritime	Page: 1 of 3
Progress report	Period/week(s): 8/2	Number of hours this period. (from log) Approx. 190	Project group (name): Group 2	Date: 22.04.2022

Main goal/purpose for this periods work
- OPC UA communication and logger - Making and testing new thruster types
Planned activities this period
- Finish python logger - Finish the OPC setup - Test and add CAN-bus to the system with an ESP-32 - Make model thruster type 1 - Test thruster type 1 - Make model thruster type 2
Actually conducted activities this period
- Creating the new thruster dynamics model in Matlab/Simulink and adding model for CPP. - Adjusting parameters and tuning the thruster dynamics / CPP model. - Planning reorganisation of report structure - Testing OPC communication in 4diac and forte - Made functional python prototype
Description of/ justification for potential deviation between planned and real activities
- The CPP model needed a lot more work than anticipated. It is also a task that blocks progress for most of the tasks for making the new thruster models in 4diac. When the other possible tasks relating to other topics were completed, the total progress was slowed because of this. - Waiting for ordered parts for the CAN-bus solution. Were told there was equivalent equipment at campus which could be borrowed until arrival of new parts, but the parts were not found.
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report
- A2 is reordered and extended to account for the current deviation. A new task A29 was made to extract the azimuth functionality from the other thruster model tasks. Thus, making this a common task to include the modelling of azimuth, programming of the ESP-32 and incorporate this in the other models. - To be able to complete the project within the timeframe, the scope had to be adjusted. It was decided that task A33 and A34 had to be removed. This is not a major disadvantage for the project since the current model is already improved. The rest of the tasks in A3 is moved 10 days to start on May 6th. - The updated Gantt-chart is shown in Figure 1.
Main experience from this period
- Propeller dynamics research, pitch dynamics, PID tuning - How mathematically pitch and water dynamics affect propeller rpm, thrust, and torque. - How Docker can be used to make a containerised environment to run forte with external libraries on a Wago PLC. - How to setup and use the internal communication between different forte runtimes, and how they exchange information.
Main purpose/focus next period
- Making and testing new thruster types - Creating solution for CAN-bus feedback to Mcon

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 2 of 3
Progress report	Period/week(s) 8/2	Number of hours this period. (from log) Approx. 190	Project group (name) Group 2	Date 22.04.2022

- Azimuth handling	
Planned activities next period - Test and add CAN-bus to the system with an ESP-32 - Azimuth dynamics - Test the OPC communication using Docker with internal communication - Make model thruster type 1 - Make model thruster type 2	
Other	
Wish/need for counselling - Nothing in particular.	
Approval/signature group leader Ole-Andre Aurvåglid Ramstad	Signature other group participants Espen Brandtzæg Godø Mari Ellinor Longva Haram

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 3 of 3
Progress report	Period/week(s) 8/2	Number of hours this period. (from log) Approx. 190	Project group (name) Group 2	Date 22.04.2022

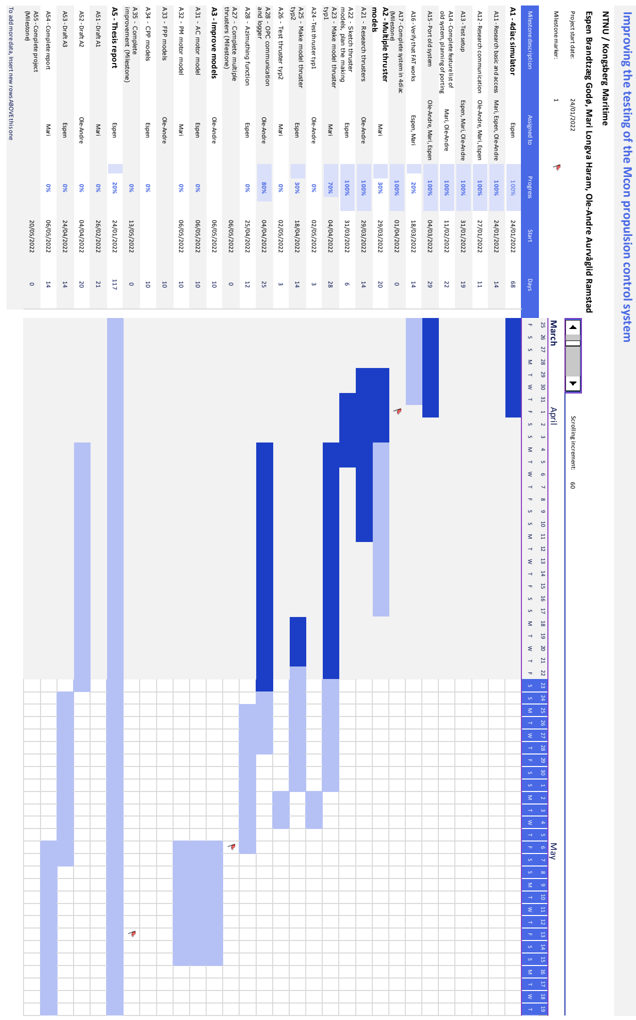


Figure 1: Updated Gantt chart

1) 1 meeting with the Steering group and 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 1 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 1 of 2
Progress report	Period/week(s) 9/1	Number of hours this period. (from log) Approx. 135	Project group (name) Group 2	Date 29.04.2022

Main goal/purpose for this periods work

- Making and testing new thruster types
- Creating solution for CAN-bus feedback to Mcon

Planned activities this period

- Test and add CAN-bus to the system with an ESP-32.
- Azimuth dynamics.
- Test the OPC communication using Docker with internal communication.
- Make model thruster type 1.
- Make model thruster type 2.

Actually conducted activities this period

- Testing of Docker with communication between multiple instances of forte and with OPC.
- Made python program to graph values.
- Discussed and decided the solution for CAN.
- Making program for the thruster type 1 and 2.
- Make model for azimuth dynamics.

Description of/ justification for potential deviation between planned and real activities

- There was a problem with the order of the parts for the CAN integration, and the CANopen is a complex and intricate protocol that will be resource intensive to implement.
- Due to sickness in the group some tasks are slightly delayed in progress but should be on track to be completed within the timeframe.

Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report

- Because of the parts needed to implement CAN communication and the complexity of integrating a CANopen node, it was decided to remove this from the project and rather simulate the feedback from the azimuth function with an analog signal.

Main experience from this period

- How to make Dockerfiles and the syntax needed to describe and setup a docker container.
- Better knowledge in how variables are used and accessed in a bash-script, and how to implement regex verification of variables.
- How different integration methods (discrete) in Simulink can significantly change the result of the model which may cause the need for design changes to achieve a similar response to the continuous model.
- How docker works contrary to a virtual machine.
- CANopen and its technicalities.

Main purpose/focus next period

- Finish thruster model 1 and 2.
- Start modelling of AC motor model.

Planned activities next period

- Add and wire up additional I/O.
- Test thruster model 1 and 2.
- Testing of azimuth function.
- Start researching and modelling of the AC motor model.

Other

Wish/need for counselling

- Nothing.

1) 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 1 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 2 of 2
Progress report	Period/week(s) 9/1	Number of hours this period. (from log) Approx. 135	Project group (name) Group 2	Date 29.04.2022

Approval/signature group leader	Signature other group participants
Espen Brandtzæg Godø	Ole-Andre Aurvåglid Ramstad

1) 1 internal meeting.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 1 of 2
Progress report	Period/week(s) 10/1	Number of hours this period. (from log) Approx. 130	Project group (name) Group 2	Date 06.05.2022

Main goal/purpose for this periods work
- Finish thruster model 1 and 2. - Start modelling of AC motor model.
Planned activities this period
- Add and wire up additional I/O. - Test thruster model 1 and 2. - Testing of azimuth function. - Start researching and modelling of the AC motor model.
Actually conducted activities this period
- Add and wire up additional I/O. - Finish system for thruster model 2 (azipull) in 4diac. - Test thruster model 2. - Testing of azimuth function. - Design and built RC-circuit. - Started making system for thruster model 1 (swing-up) in 4diac. This is an expansion of thruster model 2.
Description of/ justification for potential deviation between planned and real activities
- There was a couple of bugs in the thruster dynamics that caused the system to be unstable when the pitch angle approached zero and therefore the model had to be retuned and discretised again. - One of the tasks scheduled was removed from the plan.
Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report
- Due to time limitation and added tasks the 3rd phase was removed from the plan. The disadvantage of this is only less accurate motor dynamics since the CPP and FPP models is already improved compared to the last year's model. It is important to have enough time for gathering results and writing the final thesis. The updated Gnatt chart is shown in Figure 1.
Main experience from this period
- The time step is critical when discretising and if it is too high the model will become unstable, but this may also depend on other parameters. - How to design and make a RC-filter circuit to change a signal from PWM to 0-10V.
Main purpose/focus next period
- Make system in 4diac for last thruster type - Testing and gathering of results - Make video presentation and poster - Thesis report
Planned activities next period
- Make system in 4diac for swing-up thruster - Test swing-up on physical equipment - Gather results for azipull and swing-up - Film video presentation - Create poster - Make draft of thesis report for advisors
Other

1) 1 internal meeting and 1 meeting with the steering group.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 2 of 2
Progress report	Period/week(s) 10/1	Number of hours this period. (from log) Approx. 130	Project group (name) Group 2	Date 06.05.2022

Wish/need for counselling - Nothing.	
Approval/signature group leader Espen Brandtzæg Godø	Signature other group participants Mari Ellinor Longva Haram Ole-Andre Aurvåglid Ramstad

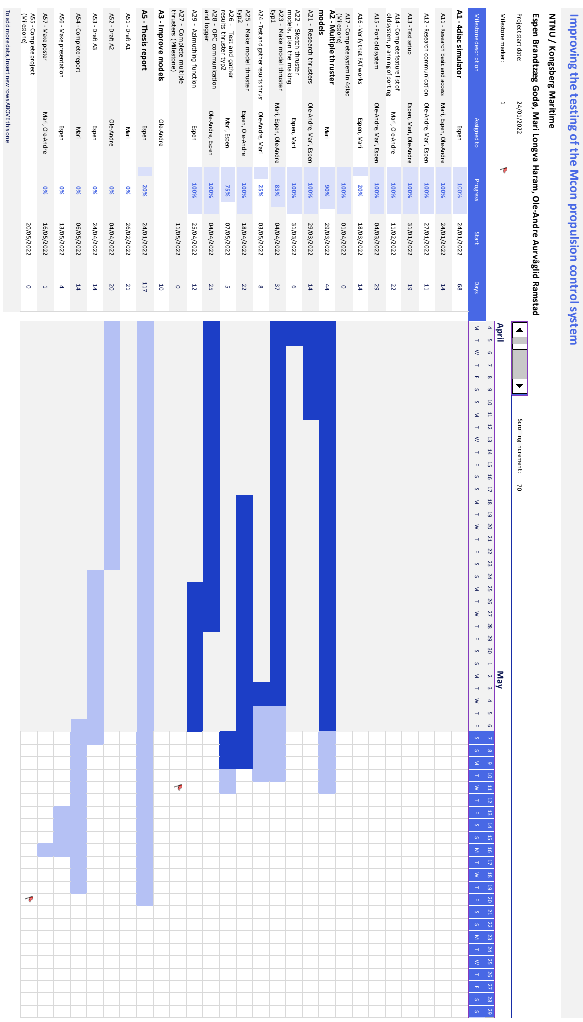


Figure 1: Updated Gantt chart

1) 1 internal meeting and 1 meeting with the steering group.

IELEA2920 Bachelor Thesis Automation	Project Improving the testing of the Mcon propulsion control system	Number of meeting this period (1). 2 planned	Company - Employer NTNU i Ålesund / Kongsberg Maritime	Page 1 of 1
Progress report	Period/week(s) 11/1	Number of hours this period. (from log) Approx. 170	Project group (name) Group 2	Date 13.05.2022

Main goal/purpose for this periods work

- Make system in 4diac for last thruster type
- Testing and gathering of results
- Make video presentation and poster
- Thesis report

Planned activities this period

- Make system in 4diac for swing-up thruster
- Test swing-up on physical equipment
- Gather results for azipull and swing-up
- Film video presentation
- Create poster
- Make draft of thesis report for advisors

Actually conducted activities this period

- Make system in 4diac for swing-up thruster
- Test swing-up on physical equipment
- Gather results for azipull and swing-up
- Make draft of thesis report for advisors

Description of/ justification for potential deviation between planned and real activities

- It took longer time than planned to gather all the results from the system
- It was decided to focus on the report for the entire week, and instead make the poster during the weekend and the filming next week. This way a new draft can be ready at an earlier time, if the advisors are available for feedback.

Description of/ justification for changes that is desired in the projects content or in the further plan of action – or progress report

- There is made no further changes to the project, the focus is now fully on the report, poster and presentation.

Main experience from this period

- Received feedback from Kongsberg about the control systems. There seems to be realistic response of the pitch and rpm control, but the azimuthing seems to be quick and could be slower.
- How to export scope data from Simulink to a high-resolution graph.

Main purpose/focus next period

-

Planned activities next period

-

Other

- This is the final progress report.

Wish/need for counselling

-

Approval/signature group leader	Signature other group participants
Mari Ellinor Longva Haram	Ole-Andre Aurvåglid Ramstad Espen Brandtzæg Godø

1) 1 internal meeting and 1 meeting with the steering group.

D Meeting minutes

Week 4: Steering group meeting

Location:	Teams
Date:	27.01.2022
Time:	13:00 – 13:30
Attendees:	Ole-Andre A. Ramstad (Project Leader), Espen B. Godø (Secretary), Mari E.L. Haram, Håkon Lunheim, Ottar L. Osen, Robin T. Bye

Agenda items

1. Project description, are there any changes that

    The steering group agrees with the description and targets of the project

2. Any changes to the preliminary report?

    There are no changes needed for the preliminary report.
    The group should divide the recourses to the tasks (at least two members per task)

3. Delivery of the test jig

    The test jig can be delivered by Kongsberg, and Project Group will contact Anders Sætersmoen (lab engineer) for placement (Tunglabben in Gnisten)

4. Communication and meetings moving forward

    Primary line of communication should be Teams
    Use phone or email if necessary
    The steering group should meet every other Thursday from 12:00 to 13:00

5. Other

    The project group have one other subject from week 4 to 11 and have reserved Monday-Wednesday for this.

    The B&R controllers should not be used in the project but will be replaced with Wago.

Action items	Owner(s)	Deadline	Status
Check who should be included in the project from KM	Håkon	10.02.22	
Check licence for B&R Automation Studio	Håkon	10.02.22	

Action items	Owner(s)	Deadline	Status
Check when the group can visit Longva, ship	Håkon		
Check with Anders Sætersmoen of placement of test jig, and access	Ole-Andre	28.01.22	
Divide members for different tasks	Ole-Andre	10.02.22	

Week 6: Steering group meeting

Location:	Teams
Date:	10.02.2022
Time:	12:00 – 12:30
Attendees:	Espen B. Godø (Project Leader), Mari E.L. Haram (Secretary), Ole-Andre A. Ramstad, Ottar L. Osen, Geir Olav Otterlei, Håkon Lunheim, Truls Antonsen

Agenda items

1. Walkthrough of the progress report from week 4

   Espen gave a walkthrough of the report. No major comments from the steering group.

2. Walkthrough of the progression of testing the system setup (Wago/Matlab)

   Wago: The group is having some challenges implementing 4diac on the Wago PLC but are still working on it. If eventually something ends up not working, Ottar recommends documenting in the report what worked and what didn't, and why.

   Matlab: The group has established connection between Matlab and 4diac. Because of changes in the Instrument Control Toolbox (specifically the tcpip class) the Matlab version used needs to be no newer than the R2020a version.

3. Available source code from last year's project?

   Håkon has the Java program and the PLC program available and will give the project group access.

4. Loading of new program to one of the screens on the test rack.

   One of the panel PCs on the test rack is running autocrossing software. Håkon will ask if Øystein or Per Magne (KM) can come and load the correct software to the panel PC.

5. Other

   Geir Olav has spoken to Einar Heggen at Longva facility about the project group visiting to see how the Mcon system is tested and get a better understanding of the project. This visit will last one day but can be expanded to two if needed. The project group is mostly available upon notice. KM will suggest date for this visit.

   KM would like to invite the project group to a visit to Ulsteinvik facility, here they can see different thrusters in real life and get a better understanding of the project. KM will suggest date for this visit.

   KM will check if they can offer the project group a walkthrough in the laboratory at NMK, lasting about half a day.

The project group should change the length of the recurring meetings with the steering group to last 30 min, as this seems to be sufficient.

Action items	Owner(s)	Deadline	Status
Make last year's source code available to the project group	Håkon	24.02.22	
Find someone to load correct software on to panel PC on the test rack	Håkon	24.02.22	
Check when the group can visit Longva, Ulsteinvik and NMK	Geir Olav		
Change meeting length to 30 min in teams/outlook	Espen	24.02.22	

Week 8: Steering group meeting

Location:	Teams
Date:	24.02.2022
Time:	12:00 – 12:35
Attendees:	Espen B. Godø, Mari E.L. Haram (Project Leader), Ole-Andre A. Ramstad(Secretary), Ottar L. Osen, Geir Olav Otterlei, Håkon Lunheim

Agenda items

1. *Walkthrough of the progress report from week 6*

   Mari gave a walkthrough of the report. Little to no comments from the steering group.

2. *Walkthrough of the progression since last time*

   Wago: The group have successfully compiled and ran the runtime environment on the wago PLC, but due to slow progress the group has decided to extend the working period on this subject to get more comfortable with the process.

   Håkon questioned about the challenges around this and is given an explanation by Espen. The explanation contained the errors when compiling and building, lacking guides and the addressing issues surrounding both the digital and analog I/O modules.

   Geir Olav complimented the groups progress regarding the runtime environment and I/O addressing issues.

3. *Other*

   Håkon informed the group about the promised ship visit is still in planning, and that they are waiting for an available moment to update the panel pc.

   Geir Olav aired the idea of the group being able to get "hands on" learning both with physical work and tuning with Per Magne at NMK.

   Ottar informed the steering group that there has open locales at NMK which the group could possibly use as their regular working area. The group accepted the transition from Gnisten to NMK should it be possible.

   Geir Olav informed that Stig Worren might have additional functions he would like added to the 4diac simulator.

   Ottar asked the group if there would be a possibility to have a "workshop" where the group would lecture on the usage of 4diac and its functions. Geir Olav chimed in and claimed interest in this workshop idea for people within Kongsberg, he suggested either a presentation or workshop.

Action items	Owner(s)	Deadline	Status
Check availability regards to transitioning to NMK	Ottar	03.03.22	

Week 10: Steering group meeting

Location:	Teams
Date:	10.03.2022
Time:	12:00 – 12:50
Attendees:	Espen B. Godø (Meeting leader), Mari E.L. Haram (Meeting secretary), Ottar L. Osen, Geir Olav Otterlei, Håkon Lunheim

Agenda items

1. *Walkthrough of the progress since last meeting*

   Espen gave a walkthrough of the activities performed since the last steering group meeting.

2. *Problem using FBs as internal FBs in other FBs*

   Håkon will check if KM has people that could be able to assist with the problems encountered in 4diac, which are mostly related to time functions and functionblocks. The group has also posted a message in 4diac's message board.

3. *Explanation for terminology and functions*

   NMEA is a communication protocol used for instance to communicate with VDR. This was made to check if the communication was functional and should not be a priority to add in 4diac.

   Clutch used to connect/disconnect parts like drive shaft, motor and generators. Ottar provided an article to show the setup of the power electrical system.

   Drive is the term for the equipment controlling the motor. This can be a frequency converter.

   Kamewa is a waterjet delivered by KM. This can be excluded in the porting of the program.

4. *Other*

   Ottar requests using colours in the Gantt chart to better show the completion status of the activities.

   Geir Olav has a case of Wago modules that the project group can borrow to use on the project.

Action items	Owner(s)	Deadline	Status
Add colours to Gantt chart to show completion status	Project group	24.03.22	

Week 12: Steering group meeting

Location:	Teams
Date:	24.03.2022
Time:	12:00 – 12:35
Attendees:	Espen B. Godø (Project leader), Mari E.L. Haram (Secretary), Ole-Andre Ramstad, Ottar L. Osen, Robin T. Bye, Geir Olav Otterlei, Håkon Lunheim, Truls Antonsen

Agenda items

1. *Update on use of internal FBs in ST in 4diac*

   There will be a new update on the nightly build of 4diac in approximately a week that should solve this issue. The group will likely test if this version is better than the current and if so, switch to the new version.

2. *Walkthrough of the progress since last meeting*

   Espen gave a walkthrough of the activities performed since the last steering group meeting. The group has made FBs for all the planned parts. The new segmentation made the porting much easier to perform. The test of the first program could possibly be finished tomorrow Friday 25.03. Håkon would maybe like to come see the project when the testing is finished.

   It is also discovered that it is possible to export MATLAB files to Structured Text that can be used in 4diac. This is probably easier than establishing communication with MATLAB.

3. *What to do with the FAT*

   There are issues with connecting the Java program for the semi-automatic FAT to the controller. Most likely the Java program was not updated when the controller was during the summer. During the meeting the project and steering group agreed that the project group should talk to Linda about it and try to get it to work, but not to spend too much time on it if the problem can't be improved.

4. *Other*

   Robin asks about progression in relation to the planned time schedule. The group is approximately two weeks behind planned schedule but moving forward the progress should be improved as the group has finished the additional subject.

   The group asks KM if they have any resources regarding thrusters as the next project period will be expanding with more thruster types. KM has a lot of documentation that can be helpful, and they will make it available to the project group.

   KM can swap out the software on the controllers when testing other thruster types later. This should not be much work.

Action items	Owner(s)	Deadline	Status
Send thruster documentation and resources to the project group	Håkon Lunheim		

Week 14: Steering group meeting

Location:	Teams
Date:	07.04.2022
Time:	12:00 – 12:30
Attendees:	Mari E.L. Haram (Project Leader), Ole-Andre A. Ramstad (Secretary), Espen B. Godø, Håkon Lunheim, Ottar L. Osen, Robin T. Bye, Truls Antonsen

Agenda items

Agenda for meeting 07.04.22

- Walkthrough of last progress rapport
- Questions from mail
  - Software/hardware for new thruster types, extra set of memory cards? Necessary to replace hardware from the backside or change GUI?
    - GUI needs to be replaced.
    - Håkon has informed Stig to provide additional information regarding the system
  - Azimuth: PWM-signal, does it exist modules which can be used in 4diac or do you have to connect a physical one?
    - Håkon strongly recommended to test the PWM-signals.
    - Ottar suggested to simulate a PWM-signal using DO-modules but informed that this is not possible.
    - An Arduino was suggested to use, Truls referred to Per Magne Dahlset as an experienced user of this method.
    - It was said that it is required to have feedback over Canbus, but this isn't possible yet in 4diac, and will have to be done through an Arduino.
  - Pitch: How to incorporate this into the system? Does it behave similarly to rpm with order and feedback?
    - It's possible, but the software needs to be altered. This will be changed.
- Other
  - Testing equipment?
    - Physical equipment was replaced by simulation on B&R

    - Since the B&R module won't be used, Arduino was suggested and Håkon confirmed it to be a valid solution.
  - Communication
    - Mari talked about the communication part of the project and the current number of workhours put into it to make it functional
      - Ottar questioned about what kind of communication and was informed of the possibility of both OPC UA and MQTT.
    - Espen explained about the problems concerning OPC UA and Forte. The cross compiler is unable to complete its compilation with OPC UA enabled.
      - Using a Raspberry Pi as a gateway was brought up and Ottar verified that this could be a functional solution
    - Ottar brought up the possibility of running a standalone OPC UA server on the Wago PLC which has access to IO and then having a standalone instance of Forte which would access the I/O through OPC
      - It was devised a plan for testing what kind of solution would be best for the tasks at hand. The group will research which of the possible solutions will work best for them, either the standalone OPC server or the Raspberry Pi gateway.
    - Robin brought up the possibility of using a docker.
      - Both Håkon, Robin and Ottar thought this to be an interesting subject, but Håkon told the group to focus on the already established task instead of changing the task to incorporate docker.

Action items	Owner(s)	Deadline	Status
Response on questions on mail	Kongsberg Maritime		Active

Week 16: Steering group meeting

Location: Teams

Date: 21.04.2022

Time: 12:00 – 12:45

Attendees: Ole-Andre A. Ramstad (Project Leader), Espen B. Godø (Secretary), Mari E.L. Haram, Håkon Lunheim, Ottar L. Osen, Geir Olav Otterlei, Stig Worren

Agenda items

Agenda for meeting 21.04.22:

- Walkthrough of last progress rapport

  Ole gave a synopsis of the progress since the last meeting. The main advances were:

  - Making of an OPC Logger application in Python, implements an OPC library
  - Making a CPP model in Matlab/Simulink
  - Setup docker on Wago to enable OPC communication

- Walkthrough of the system

  Ole explained the system setup and the main components and how we use two runtimes simultaneously on Wago. Where one is running in a docker container, and one is running natively. The group received some feedback on the diagram used and it should represent the dataflow of the system and not necessarily include ACK or request messages.

- More info about thruster data

  The group asked for more info about the characteristics of the thrusters, so more specific data can be used in the models. E.g., ramp up time, motor data from motor nameplate

  Håkon: There is no "standard" characteristics for thrusters, but for main propulsion propellers it is normal for it to accelerate at 5% per second until it reaches 80% rpm and then slow down. For a tunnel thruster the response can be faster with up to 18% per second.

  Stig: It is sufficient to estimate 8-10 seconds for the pitch response from 0 to 100 percent and 16 seconds from end to end.

- Follow-up questions about the PWM and CAN setup

  Ottar: To setup the ESP as a CANopen slave it need to have the correct address, with lower address than the main controller.

Stig: Mcon uses hardware PWM modules and for the CAN-communication follow standard CANopen setup, where the main controller has the address 127 and expects to communicate mainly with nodes addressed 21 or 22. The controller pulls data at 250Hz.

- Software update

  There must be made a new version of the GUI and controller software, Håkon will check with Per-Magne whether they can do it.

- Other
  - The presentation of the finished project will be on the 20. May, and will consist of a 15-minute presentation divided as follows:
    - 2 minutes for introduction
    - 8 minutes for presentation of the project, this is recommended to be a movie.
    - 5 minutes for questions
  - For the next meeting the timeline should be updated and presented in the next meeting. This is to help the group prioritize the remaining tasks.

Action items	Owner(s)	Deadline	Status
Make list of Can addresses	Stig Worren		Active
More data about motors	Håkon Lunheim		Active
Check up on SW status	Håkon Lunheim		Active
Update timeline	Espen Godø		Active

Week 18: Steering group meeting

Location: Teams

Date: 05.05.2022

Time: 12:00 – 12:35

Attendees: Espen B. Godø (Project Leader), Mari E.L. Haram (Secretary), Ole-Andre A. Ramstad, Ottar L. Osen, Geir Olav Otterlei, Truls Antonsen

Agenda items

Agenda for meeting 05.05.22:

- Walkthrough of progress since last meeting

  Espen gave a synopsis of the progress since the last meeting. The main advances were:

  - Made and tested dynamic model for azimuth rotation in Matlab/Simulink and 4diac
  - Made circuit for converting 24V PWM signal to 0-10V signal.
  - Made system for azipull in 4diac and started testing
  - It was decided to use an analog signal instead of the planned CAN-implementation. This was in part due to the parts not being ordered when expected, and because the task would be a big workload for this late in the project. It is also not a crucial part of the purpose of the project, so the workload was evaluated to be bigger than the reward. CAN-bus can be implemented in later improvements of the project. It should still be mentioned in the report.

- Update on the project progress with progress plan

  Espen showed the original and the current progress plan, and gave a walkthrough of which tasks are done, delayed, or removed and which are still left to do.

  Phase 2: Still need to do some development on thruster type 2, and then testing and collection of results.

  Phase 3: Because of the short amount of time left on the project, the development of AC and PM motor models are removed from the project. The improvement of the FPP and CPP models are considered completed in phase 2. This is because the CPP model from last year could not be used so a new model had to be made in phase 2. This model incorporated pitch in a better way than a scalar factor and also incorporated water dynamics.

  Other tasks: Project report, presentation/video, and poster.

  The steering group agrees with the priorities of the project group.

- Questions about the presentation

  - Norwegian or English?

    The group can decide.

  - Weighting of the presentation vs. the report

    The presentation is not officially a part of the grading, only the report is. The following activities should be included in the attachments to the report: pitch slides, poster, meeting minutes, and link to slides or video from the final presentation.

- Is the interpretation of the pitch signal correct?

  The group has interpreted the signal to be +/- 100 % pitch. Truls confirms that most of KM's thrusters work like this. Some thrusters can only adjust the pitch 0-100 %, but the project should be based on +/- 100 %.

- Suggesting a final meeting in week 19

  The project group suggests a final meeting in week 19 to present results and discuss the report draft. The steering group suggests Thursday 12.05 at 12:00.

- Other

  - The report draft can be sent to the advisors for feedback, the sooner the better. Robin suggests both sending a pdf and link to the Overleaf file. The comments will not be detail oriented.

  - Geir Olav asks if KM can contribute in some way towards the end of the project. A discussion of the produced results is suggested, and the project group will be in touch at the start of next week when more results are produced to find out when this should be. During this discussion the project group can find out if the produced results are reasonable.

  - When making the video and poster the project group should check with KM what is okay to include.

Action items	Owner(s)	Deadline	Status
Invite to teams meeting 12.05	Mari Haram	05.05.2022	Active
Send report draft to advisors	Mari Haram		Active
Contact KM for discussion of results	Mari Haram		Active

Week 19: Steering group meeting

Location:	Teams
Date:	12.05.2022
Time:	12:00 – 12:55
Attendees:	Mari E.L. Haram (Project Leader), Ole-Andre A. Ramstad (Secretary), Espen B. Godø, Håkon Lunheim, Ottar L. Osen, Truls Antonsen, Geir Olav Otterlei

Agenda items

Agenda for meeting 12.05.22

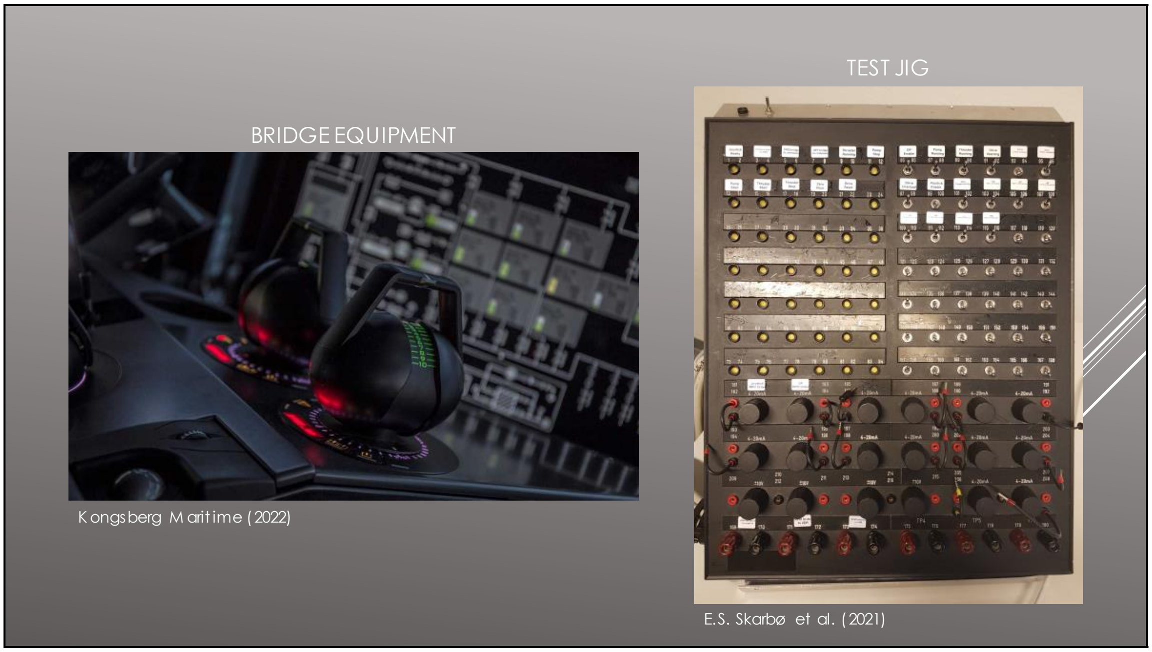
- Collected results
  - It was questioned on why the step response gave a weird graph.
    - It was explained that this was due to the sampling rate on a curved function.
  - Håkon explained that some of the models were acting fine for a simulation, but not how it actually would be. In a real setting the RPM and pitch would not both be changed, it would only be one of them, and then the other would be static.
  - He also elaborated that it would probably be better to keep the RPM static and make the pitch dynamic, which is opposite of the current setup.
  - It was also remarked that the azimuthing was a bit too fast compared to reality.
  - Håkon and Geir Olav complimented about the solution being modular and configurable
- Feedback on report draft
  - Ottar remarked that when comparing results having the same range x-axis is a must.
  - Ottar emphasized the importance of explaining why the results are the way they are.

Action items	Owner(s)	Deadline	Status

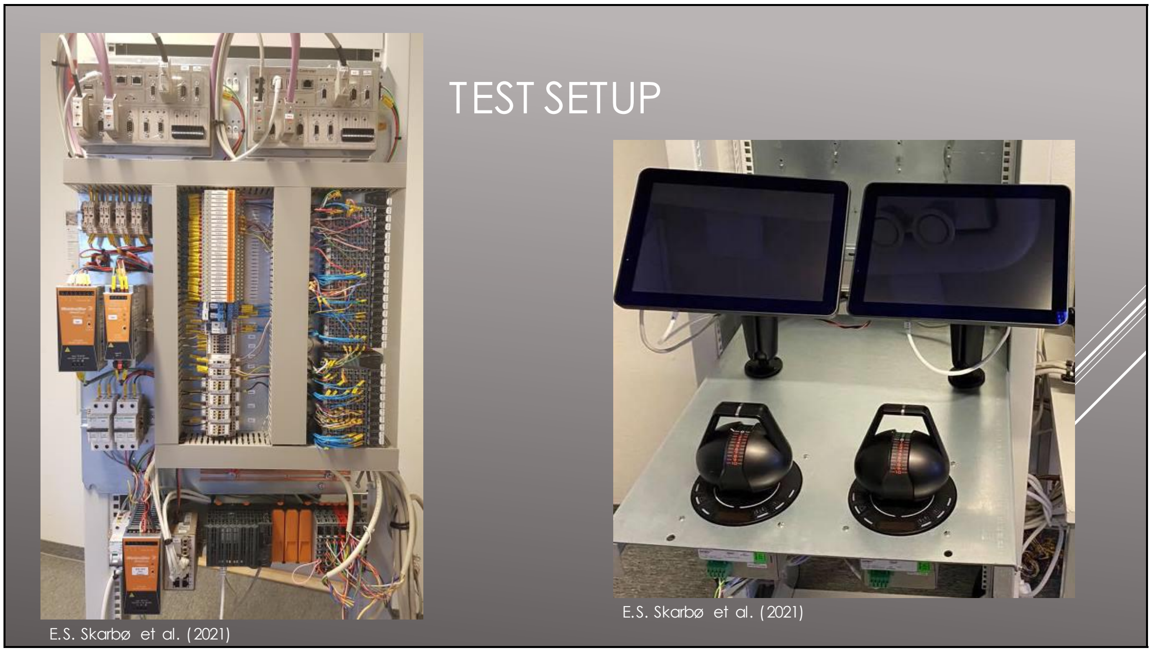
E Project pitch

IMPROVING THE TESTING OF THE MCON PROPULSION CONTROL SYSTEM

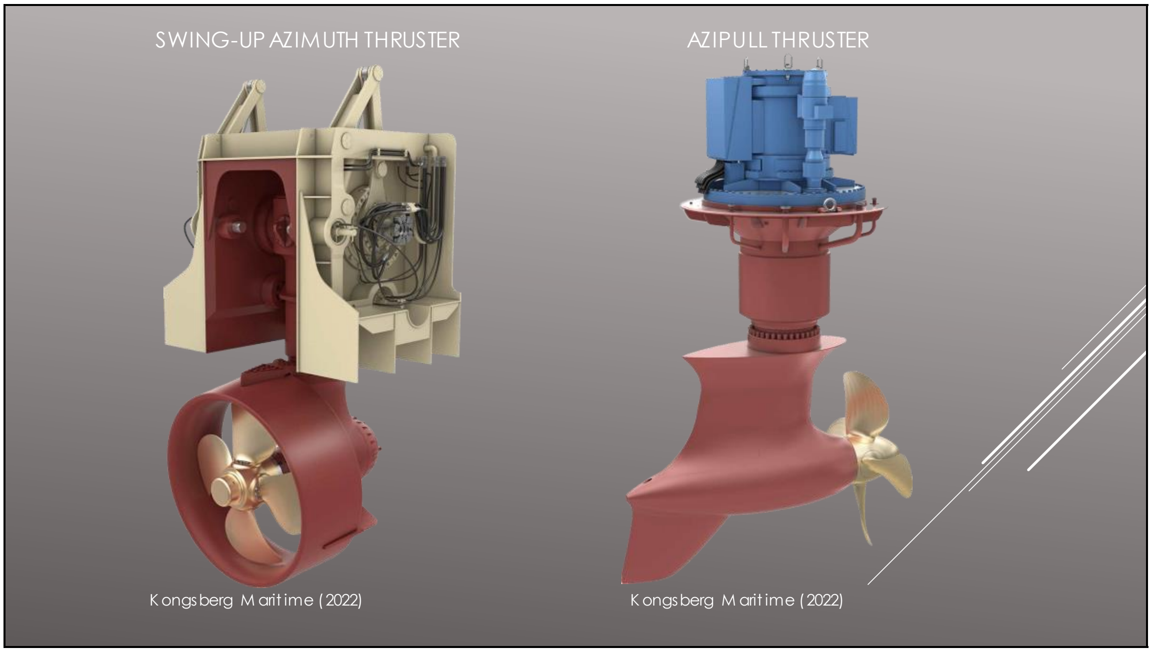
Espen Brandtzæg Godø
Mari Longva Haram
Ole-Andre Ramstad



- Streamline testing of Mcon, propulsion control system
- Semi-automate the Factory Acceptance Test procedure
- Replace the test jig with sotfware



- a thruster simulator was created in 2021 for a tunnel thruster, with a Java program for parts of the FAT
- Our goal is to extend the functionality of the simulator



Current model is only compatible with fixed pitch propeller
We will add two thruster types, the Swing-up Azimuth thruster and the Azipull thruster
Make the simulator compatible with controllable pitch propeller azimuth rotation

F Poster

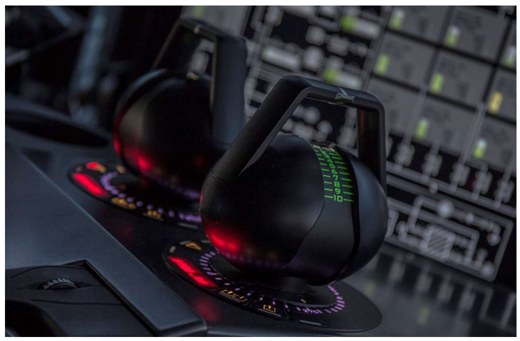
NTNU
Department of ICT and Natural Sciences

Advisors: Ottar L. Osen, Robin T. Bye
Employer: Kongsberg Maritime

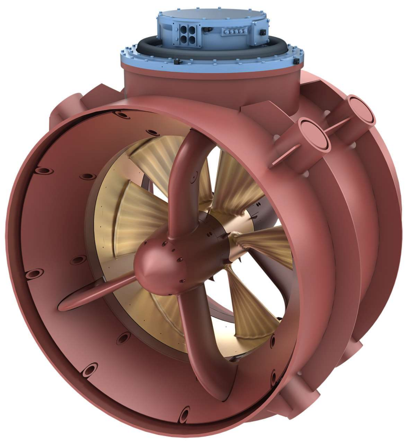
Thruster Simulator

for testing Mcon propulsion control system

A thruster simulator was created to replace physical test equipment with software, to streamline the testing. This also makes it possible to semi-automate parts of the testing procedure.



Picture: Kongsberg Maritime

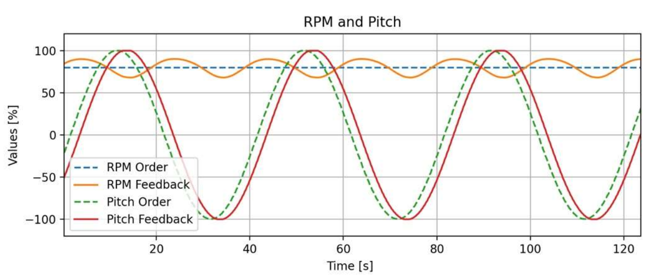


Picture: Kongsberg Maritime

A dynamic model for a thruster with adjustable pitch and azimuth rotation was made in Simulink and imported to Eclipse 4diac, an open-source PLC framework. The PLC also simulates other signals for testing. The Eclipse software was installed on a Wago PLC.

Conclusion:

- The dynamic model follows RPM and pitch order realistically
- Data is available through OPC UA for semi-automatic Factory Acceptance Testing



Espen Brandtzæg Godø | Mari Longva Haram | Ole-Andre Ramstad

G OPC address list

Below are the OPC address list that can be used to communicate with the thruster simulator.

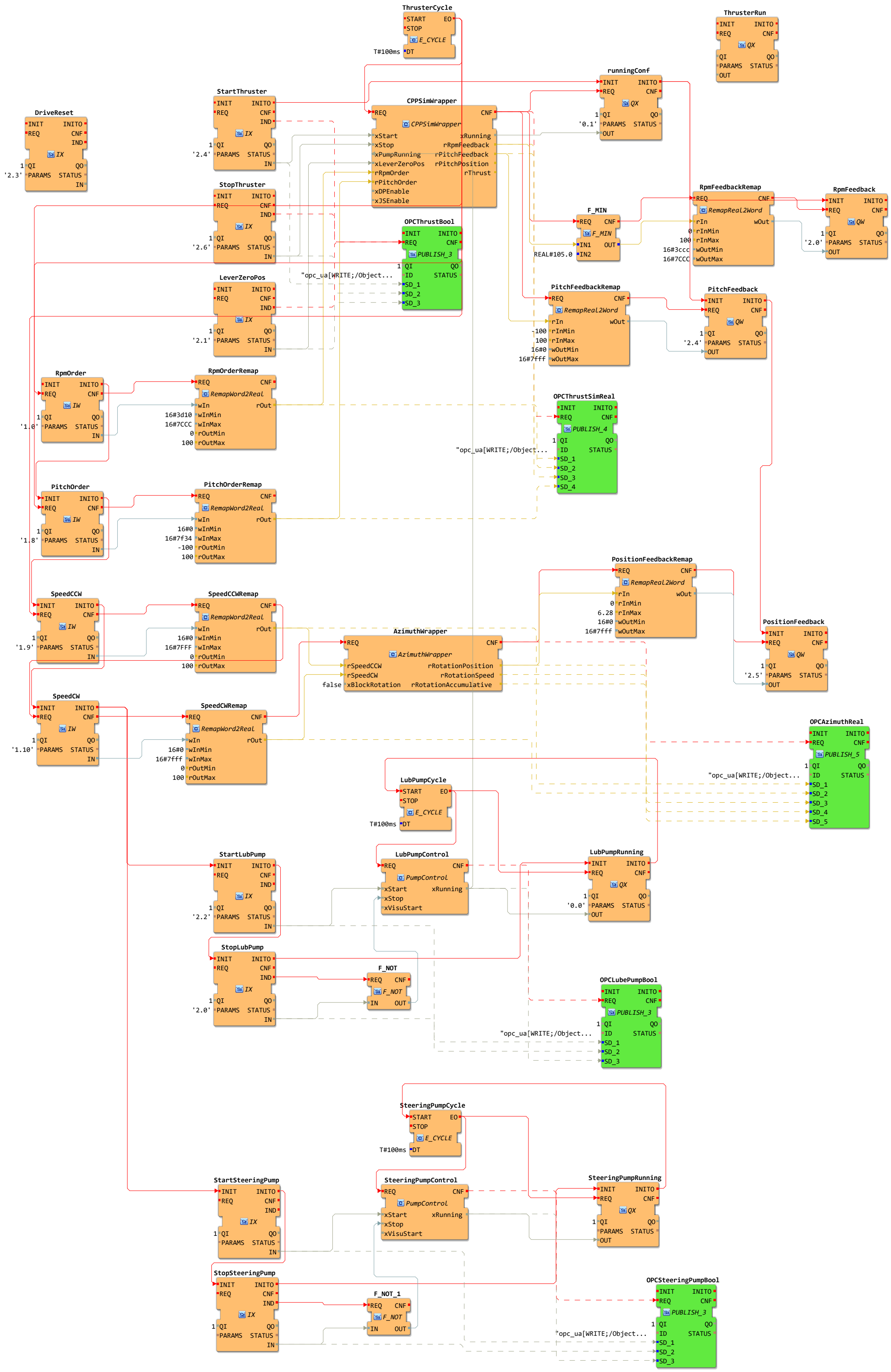
ID	Name	Directory	Function	Description
100	RpmFeedback	/Objects/ThrusterSim/	READ	The rpm feedback from the thruster simulator
101	RpmOrder	/Objects/ThrusterSim/	READ	The current rpm order from the thruster simulator [%]
102	PitchFeedback	/Objects/ThrusterSim/	READ	The pitch feedback from the thruster simulator [%]
103	PitchOrder	/Objects/ThrusterSim/	READ	The current pitch order from the thruster simulator [%]
104	AzimuthCCWOrder	/Objects/ThrusterSim/	READ	The current azimuth order in counterclockwise direction [%]
105	AzimuthCWOrder	/Objects/ThrusterSim/	READ	The current azimuth order in clockwise direction [%]
106	AzimuthPosition	/Objects/ThrusterSim/	READ	The current azimuth angular position limited to 0 - 2pi [rad]
107	AzimuthSpeed	/Objects/ThrusterSim/	READ	The current azimuth speed in [%]
108	AzimuthAccumulative	/Objects/ThrusterSim/	READ	The azimuth accumulative angular position [rad]
109	Thrust	/Objects/ThrusterSim/	READ	Calculated thrust from the thruster simulator [N]
110	Start	/Objects/ThrusterSim/	READ	Start signal
111	Stop	/Objects/ThrusterSim/	READ	Stop signal
112	LerverZeroPosition	/Objects/ThrusterSim/	READ	Lever is in zero position digital signal
210	Start	/Objects/LubPump/	READ	Start signal
211	Stop	/Objects/LubPump/	READ	Stop signal
212	Running	/Objects/LubPump/	READ	Running signal
300	Order	/Objects/DP/	READ	The rpm feedback to DP [%]
301	Feedback	/Objects/DP/	READ	The current rpm order to DP [%]
310	Ready	/Objects/DP/	READ	DP is in control, ready for order
311	Enable	/Objects/DP/	READ	DP is asking for control
350	SetRpmOrder	/Objects/DP/	WRITE	Set the rpm order from DP [%]
351	SetPitchOrder	/Objects/DP/	WRITE	Set the pitch order from DP [%]
361	SetEnable	/Objects/DP/	WRITE	Set DP to ask for control
400	Order	/Objects/JS/	READ	The rpm feedback to JS [%]
401	Feedback	/Objects/JS/	READ	The current rpm order to JS [%]
410	Ready	/Objects/JS/	READ	JS is in control, ready for order
411	Enable	/Objects/JS/	READ	JS is asking for control
450	SetRpmOrder	/Objects/JS/	WRITE	Set the rpm order from JS [%]
451	SetPitchOrder	/Objects/JS/	WRITE	Set the pitch order from JS [%]
461	SetEnable	/Objects/JS/	WRITE	Set JS to ask for control
500	Alarm	/Objects/Visu/	READ	Mcon alarm active
501	FwdControl	/Objects/Visu/	READ	The forward bridge is in control
502	AftControl	/Objects/Visu/	READ	The Aft bridge is in control
503	ThrusterRunning	/Objects/Visu/	READ	Thruster running signal
600	Start	/Objects/SteeringPump/	READ	Start signal
601	Stop	/Objects/SteeringPump/	READ	Stop signal
602	Running	/Objects/SteeringPump/	READ	Running signal

H Signal list

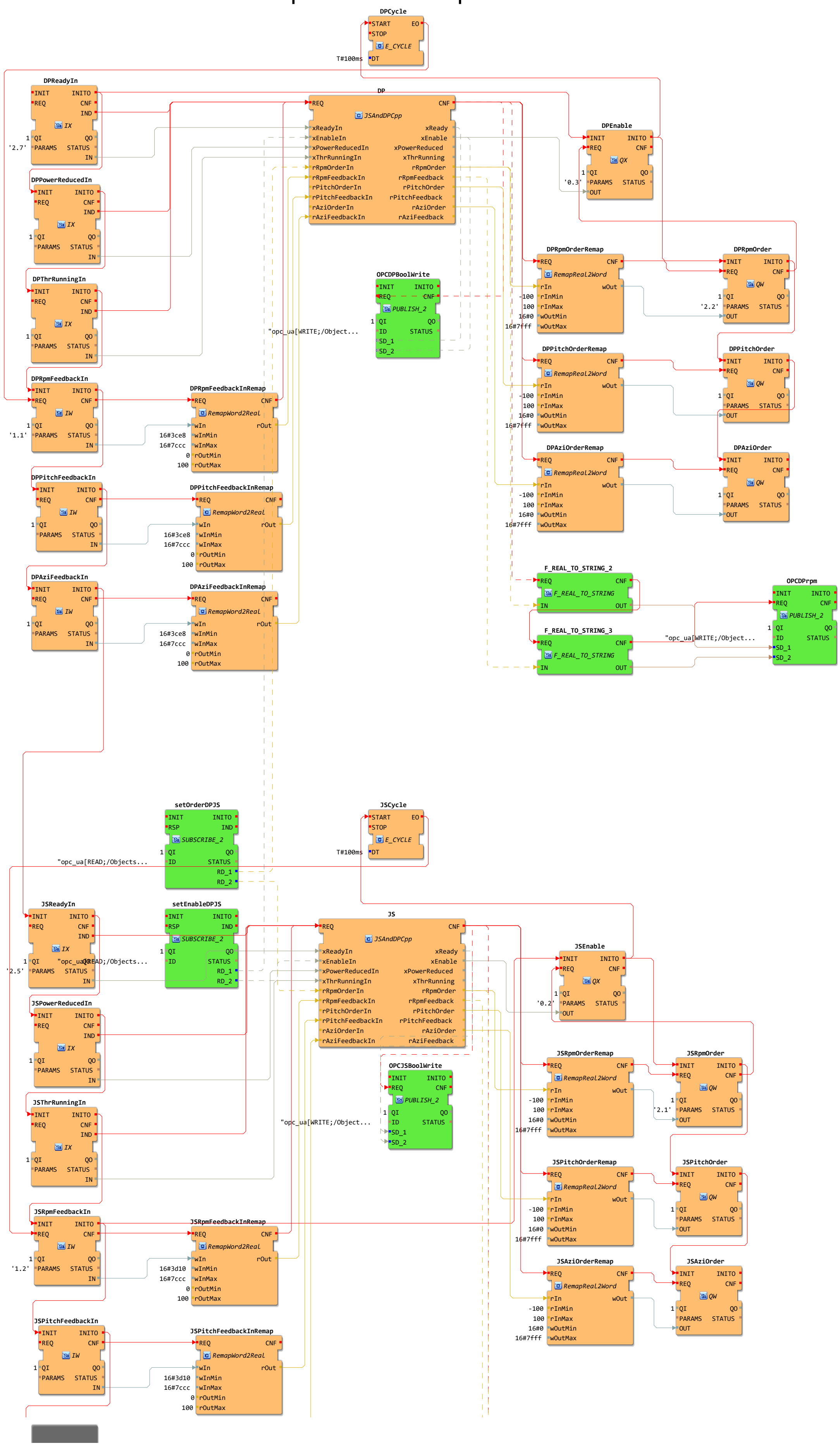
		Mcon			Wago PLC					Test Jigg	
Signal	Note	Termination	Connection	Potential	Termination	Connection	Potential	Purpose in 4diac	Address in4diac	Connection	Potential
Pump running	(closed=running)	NGE3	21	Di	A201	1	NO	Pump sim	0.0		
Pump running		NGE3	24	24V	A201	2	C	Pump sim			
Stop system pump	(open=stop)	X214	3	NC	A204	1	Di	Pump sim	2.0		
Common		X214	4	C	A204	-	24V	Pump sim			
Common		X214	2	C	A204	-	24V	Pump sim			
Start System Pump	(closed=start)	X214	1	NO	A204	2	Di	Pump sim	2.2		
RPM Order	4-20mA	X212	1	Ao	A208	1	Ai	Thr sim	1.0		
RPM Order		X212	2	Ref	A208	2	0V	Thr sim			
Drive RPM Feedback	4-20mA	NGE5	11	Ai	A206	1	Ao	Thr sim	2.0		
Drive RPM Feedback		NGE5	13	0V	A206	3	Ref	Thr sim			
Thruster Start	(closed=start)	KNGE2.1	14	NO	A204	3	Di	Thr sim	2.4		
Thruster Start		KNGE2.1	11	C	A204	-	24V	Thr sim			
Thruster Stop	(closed=stop)	KNGE2.2	14	NO	A204	4	Di	Thr sim	2.6		
Thruster Stop		KNGE2.2	11	C	A204	-	24V	Thr sim			
Zero pitch/lever in zero position	(closed=zero)	KNGE2.6	14	NO	A204	5	Di	Thr sim	2.1		
Zero pitch/lever in zero position		KNGE2.6	11	C	A204	-	24V	Thr sim			
Drive reset	(closed=reset)	KNGE2.3	14	NO	A204	6	Di	Thr sim	2.3		
Drive reset		KNGE2.3	11	C	A204	-	24V	Thr sim			
Thruster running	(closed=running)	NGE3	22	Di	A201	5	NO	Thr sim	0.1		
Thruster running		NGE3	25	24V	A201	6	C	Thr sim			
Joystick enable	(closed=enable)	NCC2	21	Di	A202	1	NO	JS sim	0.2		
Joystick enable		NCC2	24	24V	A202	2	C	JS sim			
Joystick ready	(closed=ready)	NJS1	11	NO	A204	7	Di	JS sim	2.5		
Joystick ready		NJS1	12	C	A204	-	24V	JS sim			
Joystick rpm order	4-20mA	NJS4	21	Ai	A206	5	Ao	JS sim	2.1		
Joystick rpm order		NJS4	23	0V	A206	7	Ref	JS sim			
Rpm feedback to joystick	4-20mA	NJS2	23	Ao	A208	3	Ai	JS sim	1.2		
Rpm feedback to joystick		NJS2	24	Ref	A208	4	0V	JS sim			
DP enable	(closed=enable)	NCC2	12	Di	A202	5	NO	DP sim	0.3		
DP enable		NCC2	15	24V	A202	6	C	DP sim			
DP ready	(closed=ready)	NDP1	11	NO	A204	8	Di	DP sim	2.7		
DP ready		NDP1	12	C	A204	-	24V	DP sim			
DP rpm order	4-20mA	NDP4	21	Ai	A207	1	Ao	DP sim	2.2		
DP rpm order		NDP4	23	0V	A207	3	Ref	DP sim			
DP rpm feedback	4-20mA	NDP2	23	Ao	A208	5	Ai	DP sim	1.1		
DP rpm feedback		NDP2	24	Ref	A208	6	0V	DP sim			
RPM Order	4-20mA	NAO5	21	Ao	A208	7	Ai	Visualisation	1.3(not in use)		
RPM Order		NAO5	22	Ref	A208	8	0V	Visualisation			
Common Alarm	(closed=alarm)	KNDO5.4	14	NO	A205	1	Di	Visualisation	2.8		
Common Alarm		KNDO5.4	11	C	A205	-	24V	Visualisation			
FWD Bridge In Command	(closed=command)	KNDO5.1	14	NO	A205	2	Di	Visualisation	2.10		
FWD Bridge In Command		KNDO5.1	11	C	A205	-	24V	Visualisation			
AFT Bridge In Command	(closed=command)	KNDO5.2	14	NO	A205	3	Di	Visualisation	2.12		
AFT Bridge In Command		KNDO5.2	11	C	A205	-	24V	Visualisation			
Thruster Running	(closed=running)	KNDO5.3	14	NO	A205	4	Di	Visualisation	2.14		
Thruster running		KNDO5.3	11	C	A205	-	24V	Visualisation			
RPM Feedback	4-20mA	NAO5	23	Ao	A209	1	Ai	Visualisation	1.4(not in use)		
RPM Feedback		NAO5	24	Ref	A209	2	0V	Visualisation			
RPM Reduced	(closed=alarm)	NDR1	11	Di	A221	1	Do	Thr sim			
RPM Reduced		NDR1	14	24V	A221	2	24V	Thr sim			
RPM Local Control	(closed=alarm)	NGE3	13	Di	A221	5	Do	Thr sim			
RPM Local Control		NGE3	16	24V	A221	6	24V	Thr sim			
Drive RPM Overload	(open=alarm)	BTT2	23	Di	A222	2	NO	Thr sim			
Drive RPM Overload		BTT2	26	24V	A222	3	C	Thr sim			
Drive RPM Warning	(open=alarm)	BTT1	23	Di	A222	6	NO	Thr sim			
Drive RPM Warning		BTT1	26	24V	A222	7	C	Thr sim			
Azimuth order CCW					A210	5	Ai	Thr sim	1.9	Tj169	Ai (connected to circuit)
Azimuth order CCW					A210	6	0V	Thr sim		Tj170	0V (connected to circuit)
Azimuth order CW					A210	3	Ai	Thr sim	1.10	Tj171	Ai (connected to circuit)
Azimuth order CW					A210	4	0V	Thr sim		Tj172	0V (connected to circuit)
Azimuth Feedback					A213	5	Ao	Thr sim	2.5	Tj175	Ai
Azimuth Feedback					A213	7	Ref	Thr sim		Tj176	0V
Propeller Pitch	(0 -10 V)				A210	1	AI	Thr sim	1.8	TJ209	Ao
Propeller Pitch					A210	2	0V	Thr sim		TJ210	Ao
Pitch Feedback					A213	1	Ao	Thr sim	2.4	Tj177	Ai
Pitch Feedback					A213	3	Ref	Thr sim		Tj178	0V
Lifting Cylinder in Upper Position	(closed=upper)				A215	1	Do	Swing-up	0.4	TJ25	Di
Lifting Cylinder in Upper Position					A215	2	24V	Swing-up		TJ26	0V
Lifting Cylinder in Lower Position NC	(closed=lower)				A215	5	Do	Swing-up	0.5	TJ27	Di
Lifting Cylinder in Lower Position NC					A215	6	24V	Swing-up		TJ28	0V
Lifting Cylinder in Lower Position BC	(closed=lower)				A216	1	Do	Swing-up	0.6	TJ29	Di
Lifting Cylinder in Lower Position BC					A216	2	24V	Swing-up			0V
Upper Locking Cylinder Lock					A212	1	Di	Swing-up	2.16	TJ127	Do
Upper Locking Cylinder Unlock					A212	2	Di	Swing-up	2.18	TJ129	Do
Upper Locking Cylinder 0V								Swing-up			
Lower Locking Cylinder Lock					A212	3	Di	Swing-up	2.20	Tj131	Do
Lower Locking Cylinder Unlock					A212	4	Di	Swing-up	2.22	Tj133	Do
Lower Locking Cylinder 0V								Swing-up			
Thruster in Zero Position	(closed=zero)				A220	5	Do	Swing-up	0.15	TJ47	Di
Thruster in Zero Position					A220	6	0V	Swing-up		Tj48	0V
Lower Order					A212	5	Di	Swing-up	2.17	Tj135	Do
Lower Order								Swing-up			
Lift Order					A212	6	Di	Swing-up	2.19	TJ137	Do
Lift Order								Swing-up			
Upper Locking Bolt 1 Locked	(closed=locked)				A216	5	Do	Swing-up	0.7	TJ31	Di
Upper Locking Bolt 1 Locked					A216	6	0V	Swing-up		Tj32	24V
Upper Locking Bolt 1 Unlocked	(closed=unlocked				A217	1	Do	Swing-up	0.8	Tj33	Di
Upper Locking Bolt 1 Unlocked					A217	2	0V	Swing-up		TJ34	24V
Lower Locking Bolt 1 Locked	(closed=locked)				A217	5	Do	Swing-up	0.9	TJ35	Di
Lower Locking Bolt 1 Locked					A217	6	0V	Swing-up		TJ36	24V
Lower Locking Bolt 1 Unlocked	(closed=unlocked				A218	1	Do	Swing-up	0.10	TJ37	Di
Lower Locking Bolt 1 Unlocked					A218	2	0V	Swing-up		TJ38	24V
Upper Locking Bolt 2 Locked	(closed=locked)				A218	5	Do	Swing-up	0.11	TJ39	Di
Upper Locking Bolt 2 Locked					A218	6	0V	Swing-up		TJ40	24V
Upper Locking Bolt 2 Unlocked	(closed=unlocked				A219	1	Do	Swing-up	0.12	Tj41	Di
Upper Locking Bolt 2 Unlocked					A219	2	0V	Swing-up		Tj42	24V
Lower Locking Bolt 2 Locked	(closed=locked)				A219	5	Do	Swing-up	0.13	TJ43	Di
Lower Locking Bolt 2 Locked					A219	6	0V	Swing-up		TJ44	24V
Lower Locking Bolt 2 Unlocked	(closed=unlocked				A220	1	Do	Swing-up	0.14	TJ45	Di
Lower Locking Bolt 2 Unlocked					A220	2	0V	Swing-up		TJ46	24V

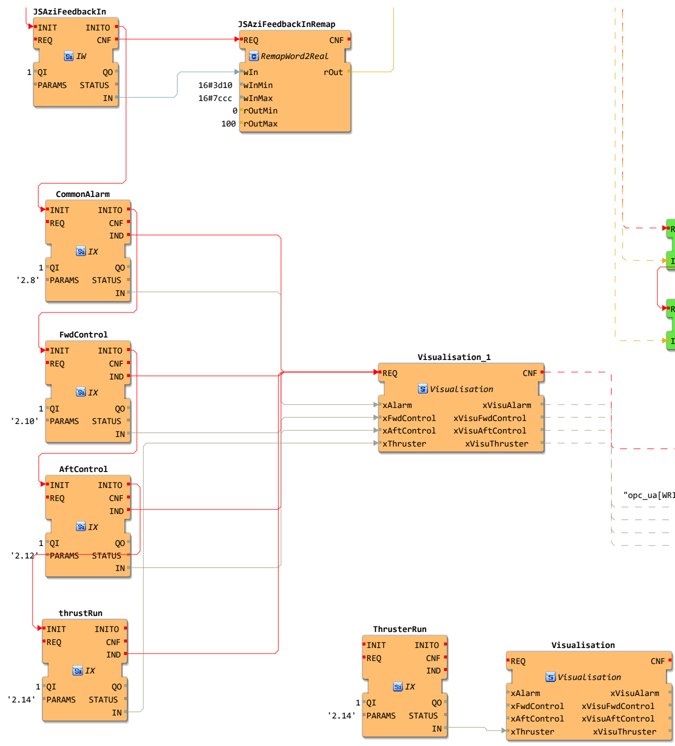
I 4diac source code

azipullSimulator thrusterSim

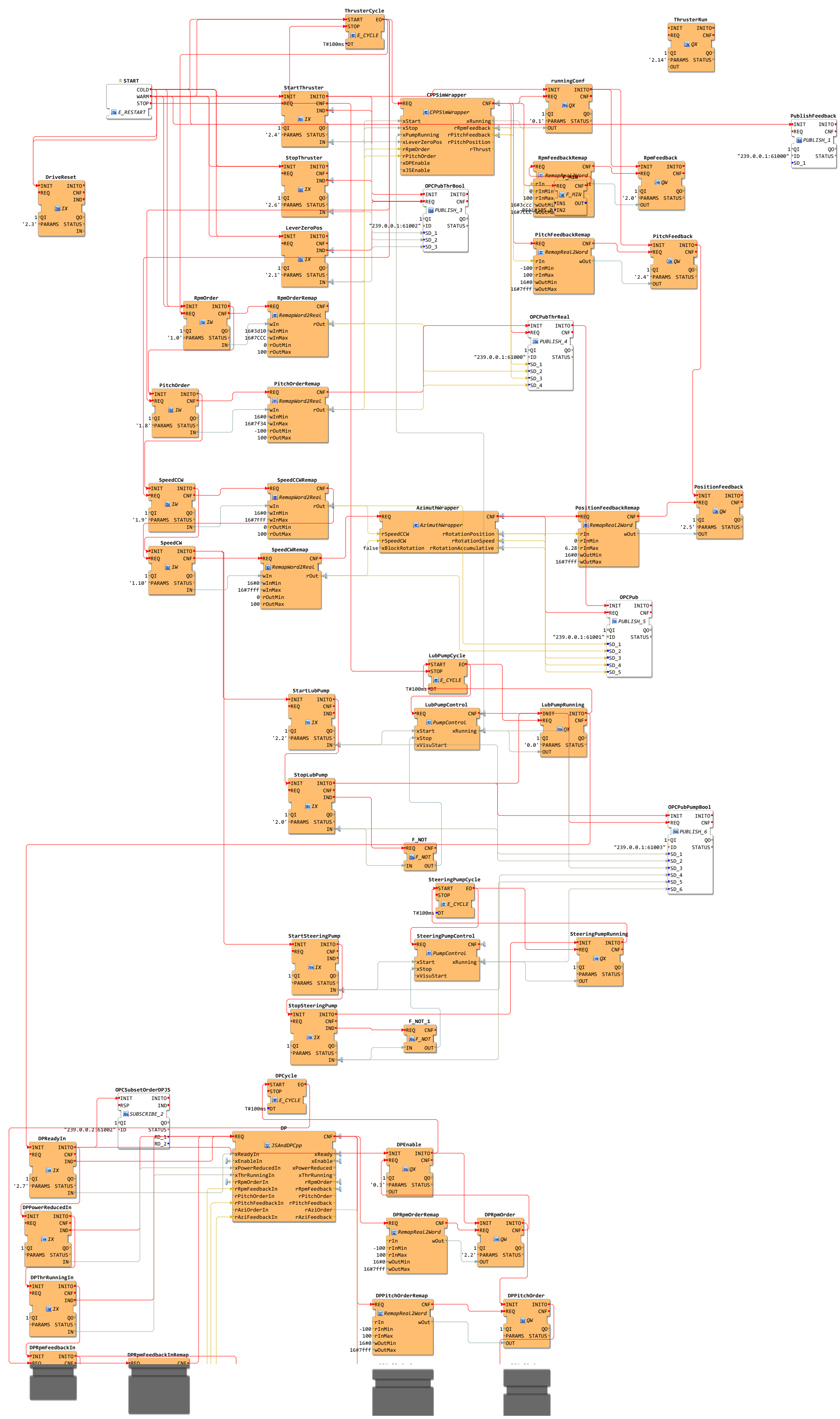


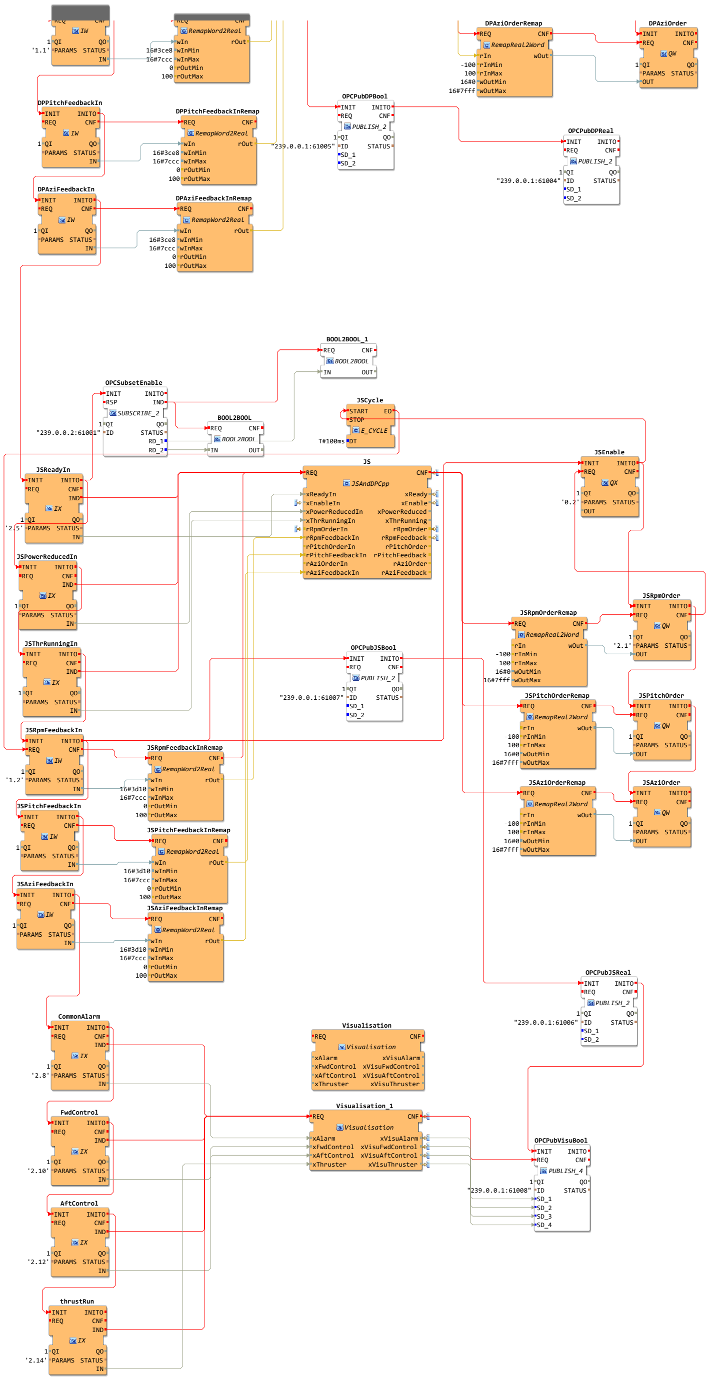
azipullSimulator dpJsVisuSim



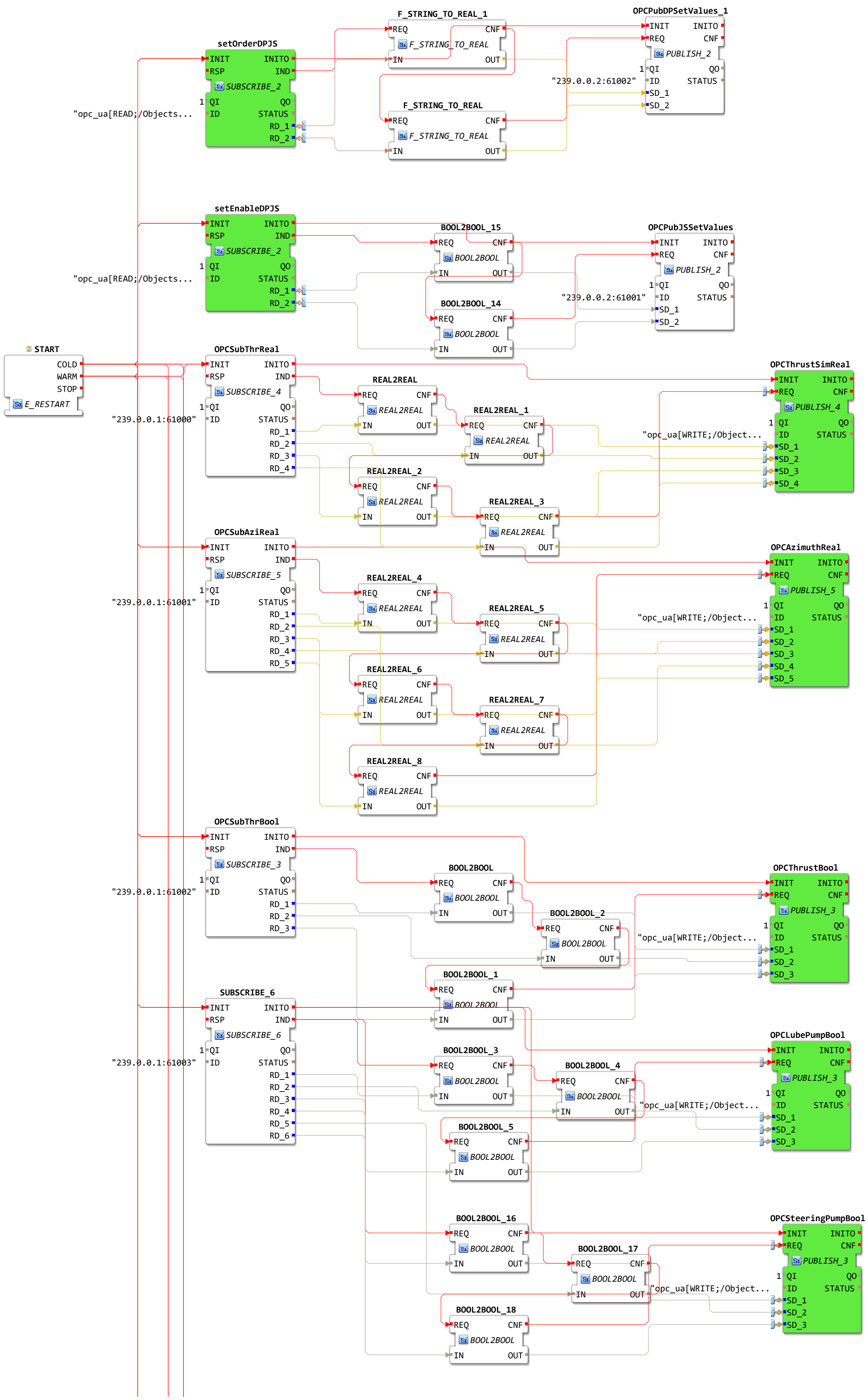

JSAziFeedbackIn
JSAziFeedbackInRemap
RemapWord2Real
16#3d10
16#7ccc
CommonAlarm
'2.8'
FwdControl
'2.10'
AftControl
'2.12'
thrustRun
'2.14'
Visualisation_1
F_REAL_TO_STRING_4
F_REAL_TO_STRING_5
OPPJSrpm
PUBLISH_2
"opc_ua[WRITE;/Object...
OPCVisuBool
PUBLISH_4
ThrusterRun
Visualisation

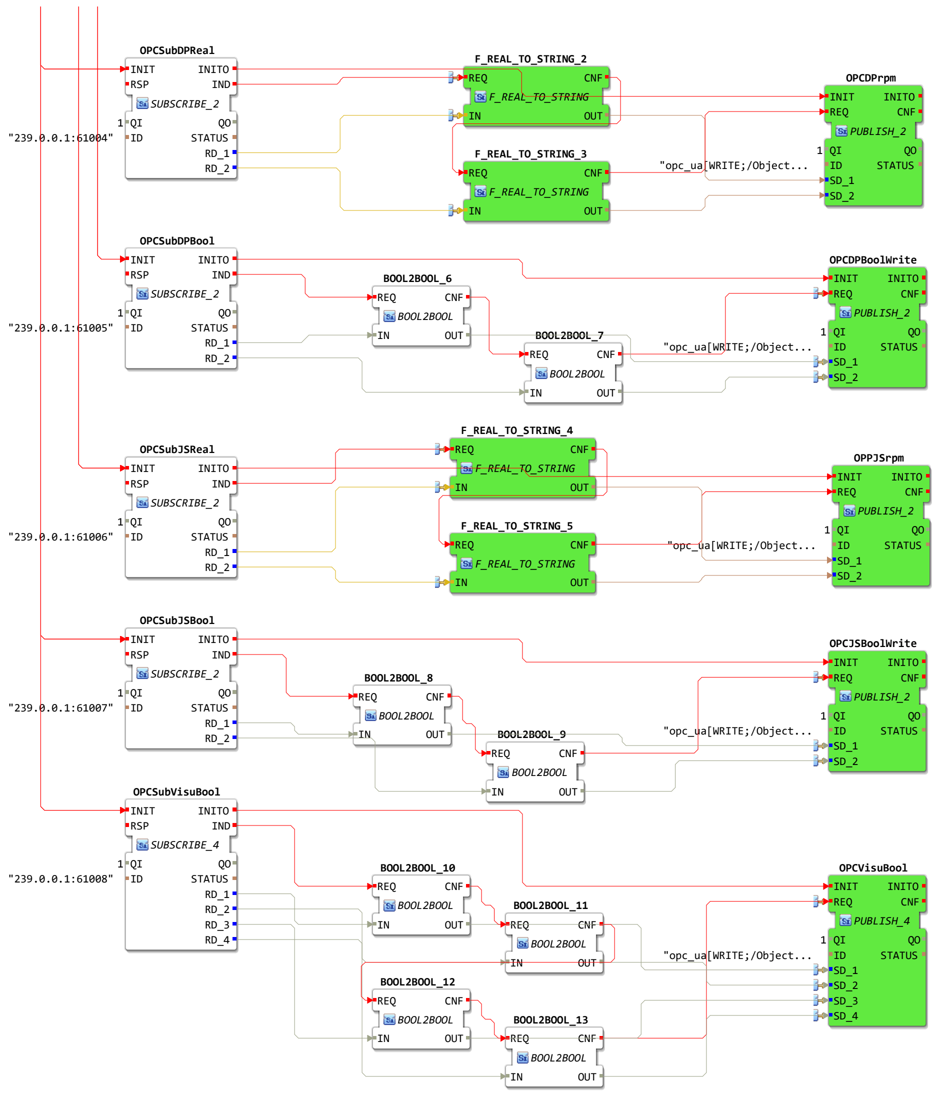
azipullSimulator A200



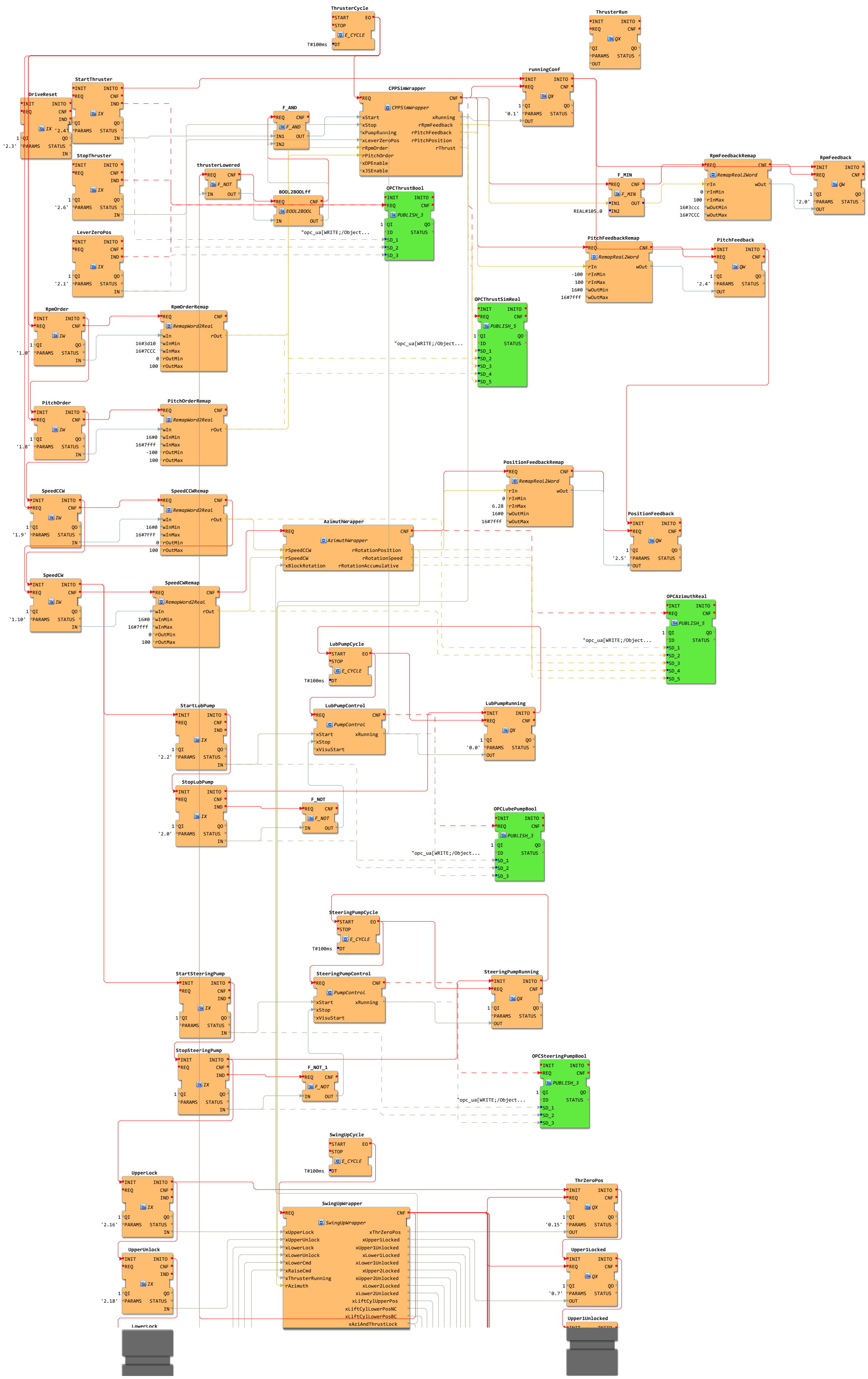


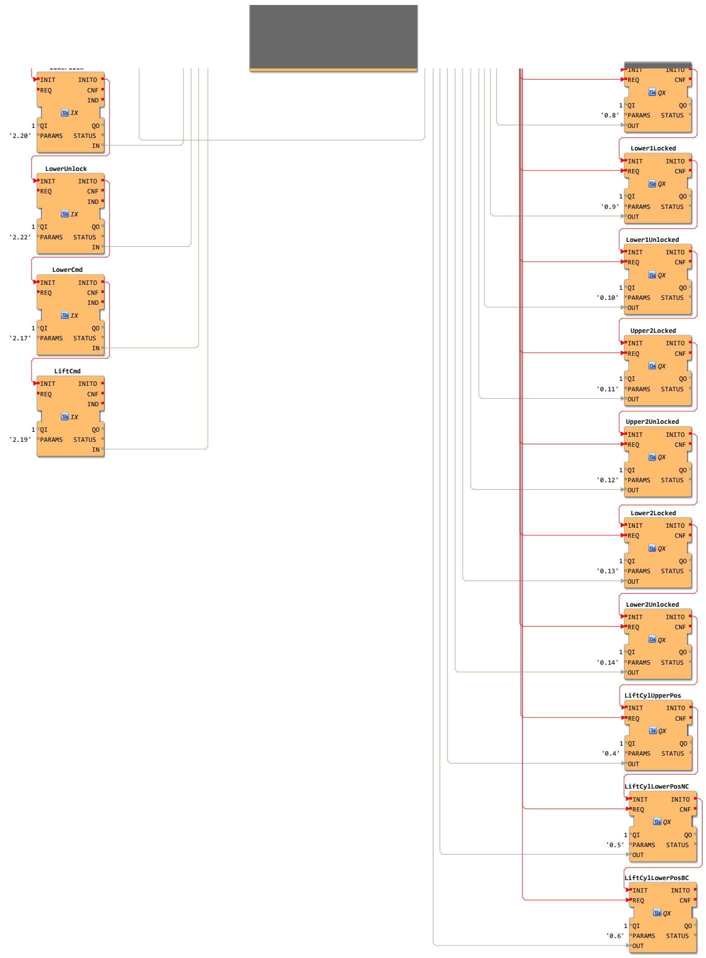
azipullSimulator A200_OPC



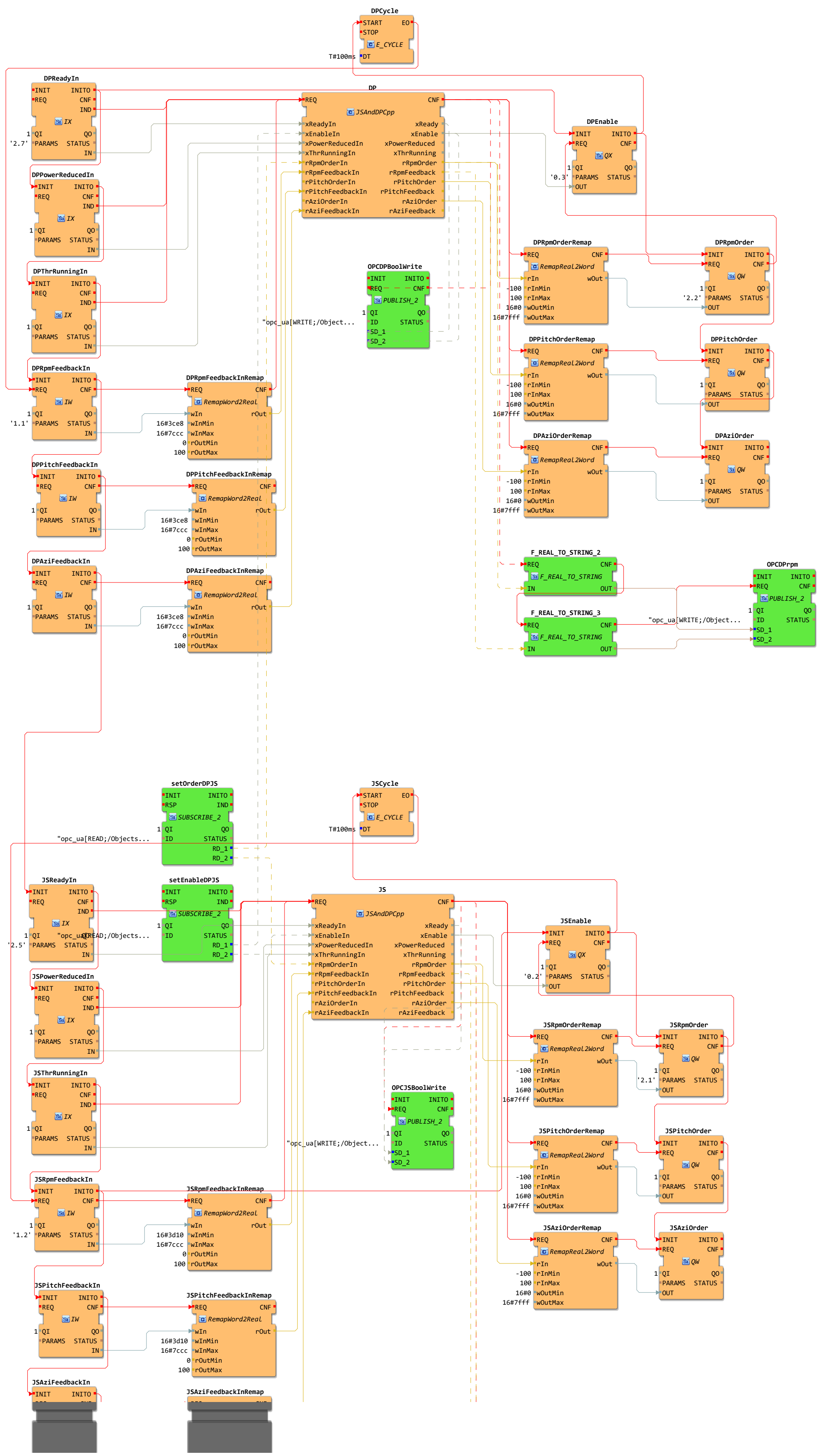

OPCSubDPReal
SUBSCRIBE_2
"239.0.0.1:61004"
F_REAL_TO_STRING_2
F_REAL_TO_STRING_3
OPCDPrpm
PUBLISH_2
"opc_ua[WRITE;/Object...
OPCSubDPBool
"239.0.0.1:61005"
BOOL2BOOL_6
BOOL2BOOL_7
OPCDPBoolWrite
OPCSubJSReal
"239.0.0.1:61006"
F_REAL_TO_STRING_4
F_REAL_TO_STRING_5
OPPJSrpm
OPCSubJSBool
"239.0.0.1:61007"
BOOL2BOOL_8
BOOL2BOOL_9
OPCJSBoolWrite
OPCSubVisuBool
SUBSCRIBE_4
"239.0.0.1:61008"
BOOL2BOOL_10
BOOL2BOOL_11
BOOL2BOOL_12
BOOL2BOOL_13
OPCVisuBool
PUBLISH_4

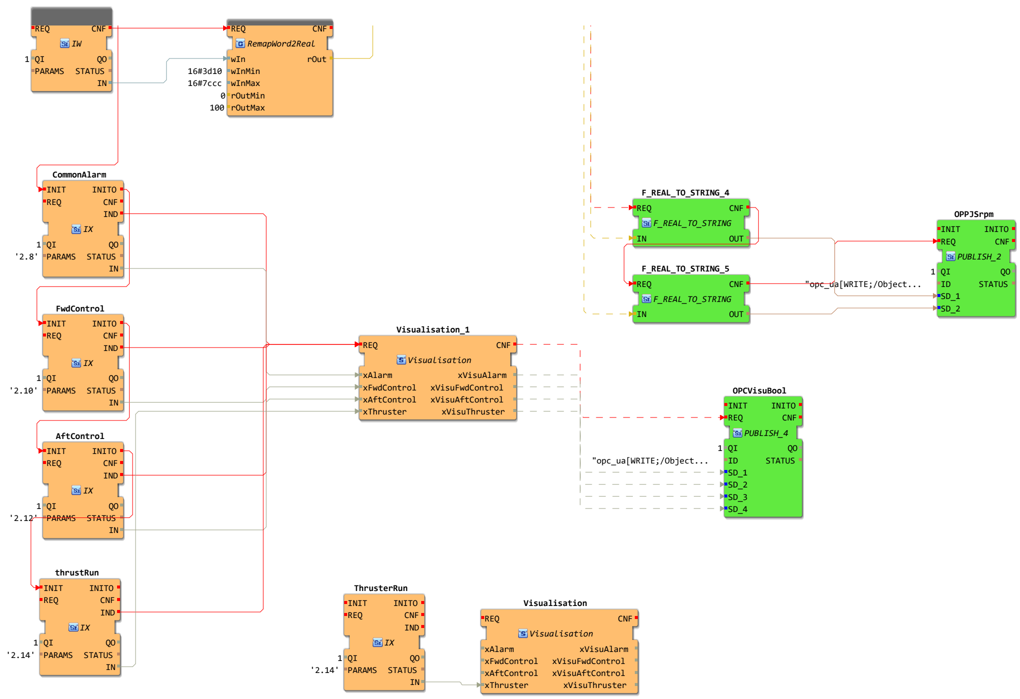
swingUpSimulator thrusterSim



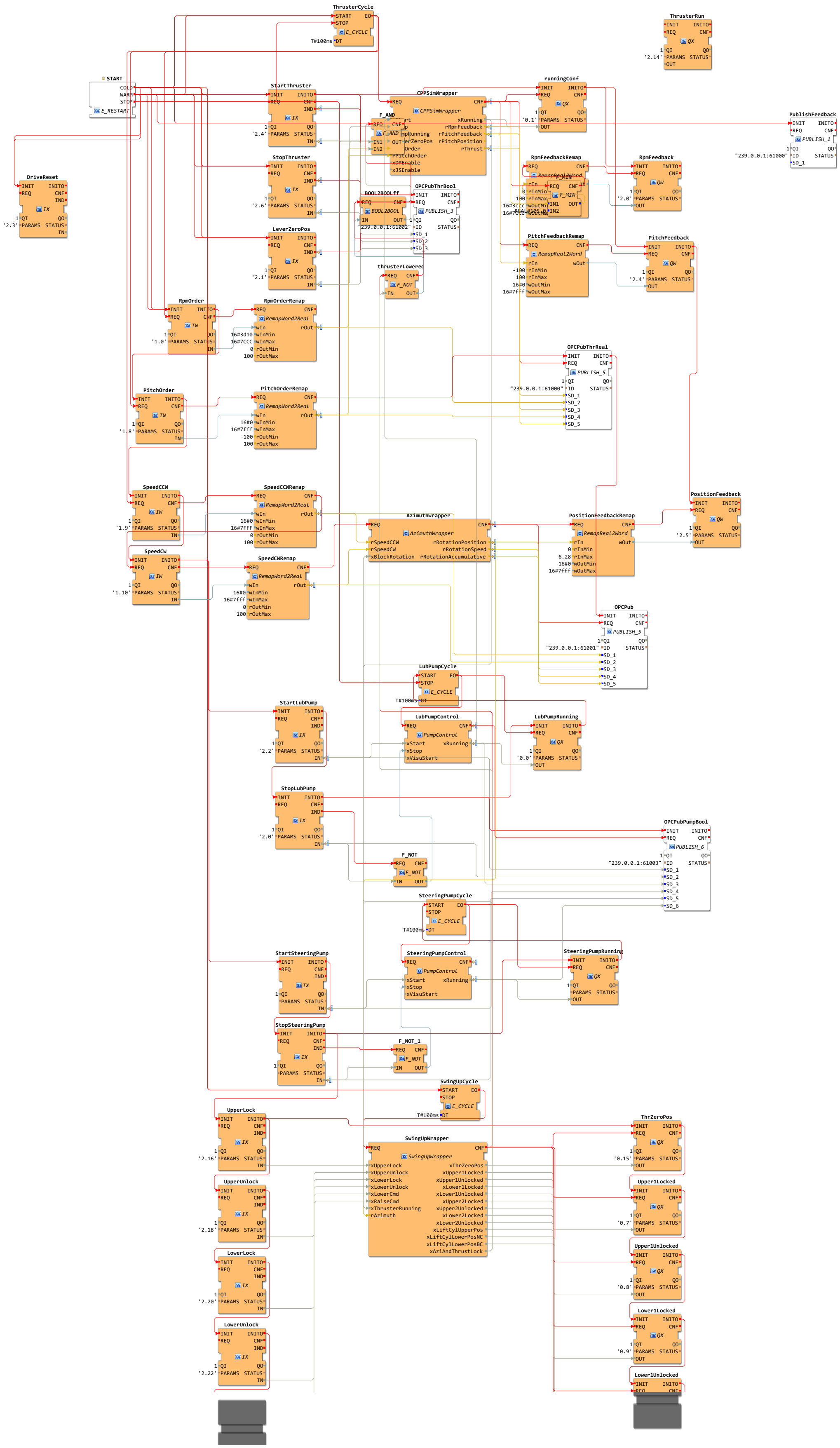

LowerUnlock
LowerCmd
LiftCmd
Lower1Locked
Lower1Unlocked
Upper2Locked
Upper2Unlocked
Lower2Locked
Lower2Unlocked
LiftCylUpperPos
LiftCylLowerPosNC
LiftCylLowerPosBC
'2.20'
'2.22'
'2.17'
'2.19'
'0.8'
'0.9'
'0.10'
'0.11'
'0.12'
'0.13'
'0.14'
'0.4'
'0.5'
'0.6'

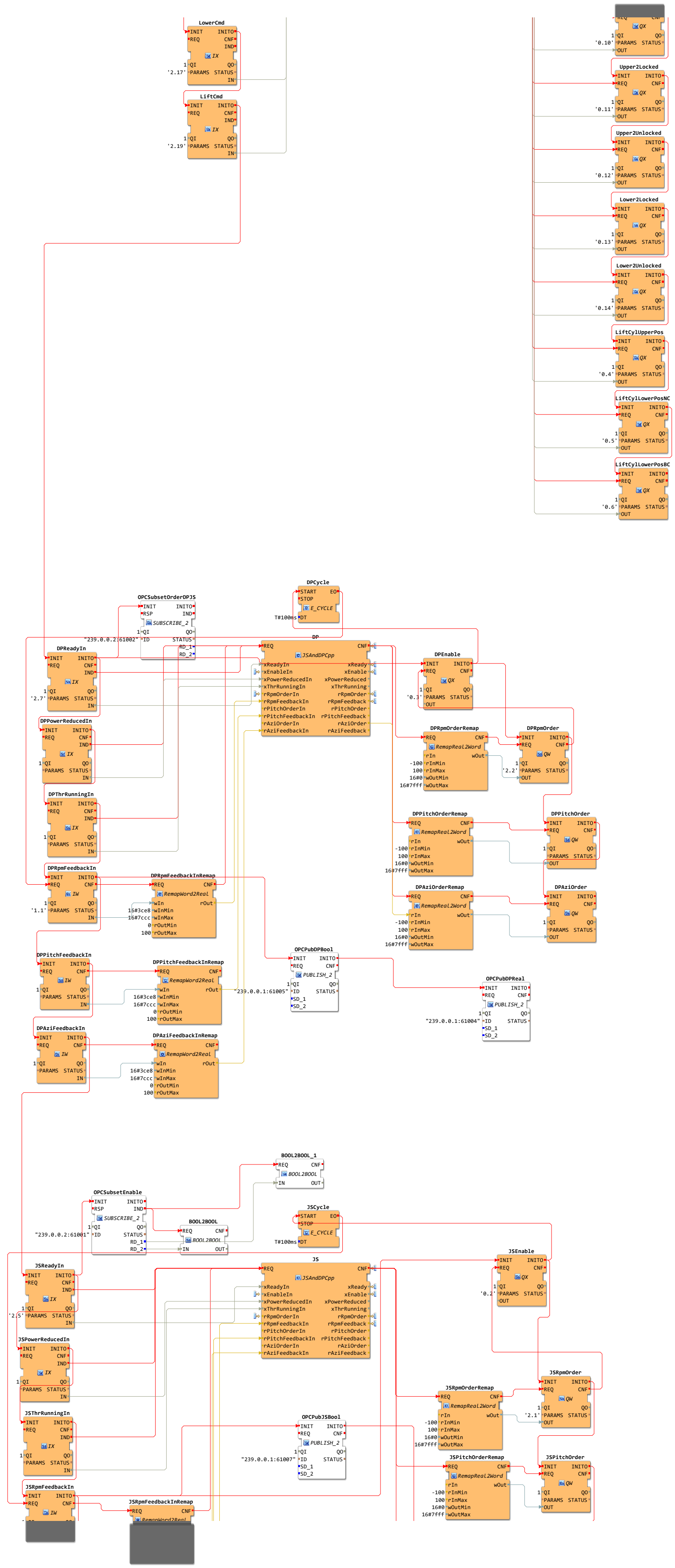
swingUpSimulator dpJsVisuSim

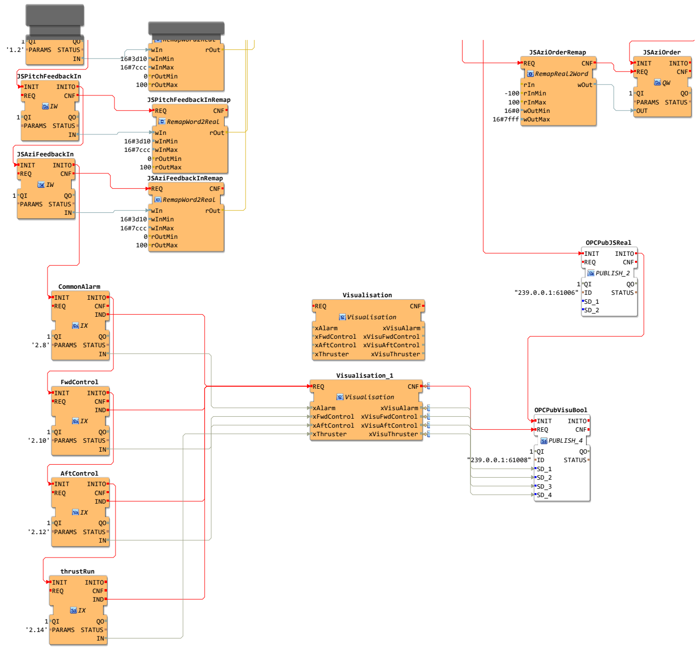



RemapWord2Real
CommonAlarm
FwdControl
AftControl
thrustRun
Visualisation_1
ThrusterRun
Visualisation
F_REAL_TO_STRING_4
F_REAL_TO_STRING_5
OPPJSrpm
OPCVisuBool
"opc_ua[WRITE;/Object...
'2.8'
'2.10'
'2.12'
'2.14'
16#3d10
16#7ccc

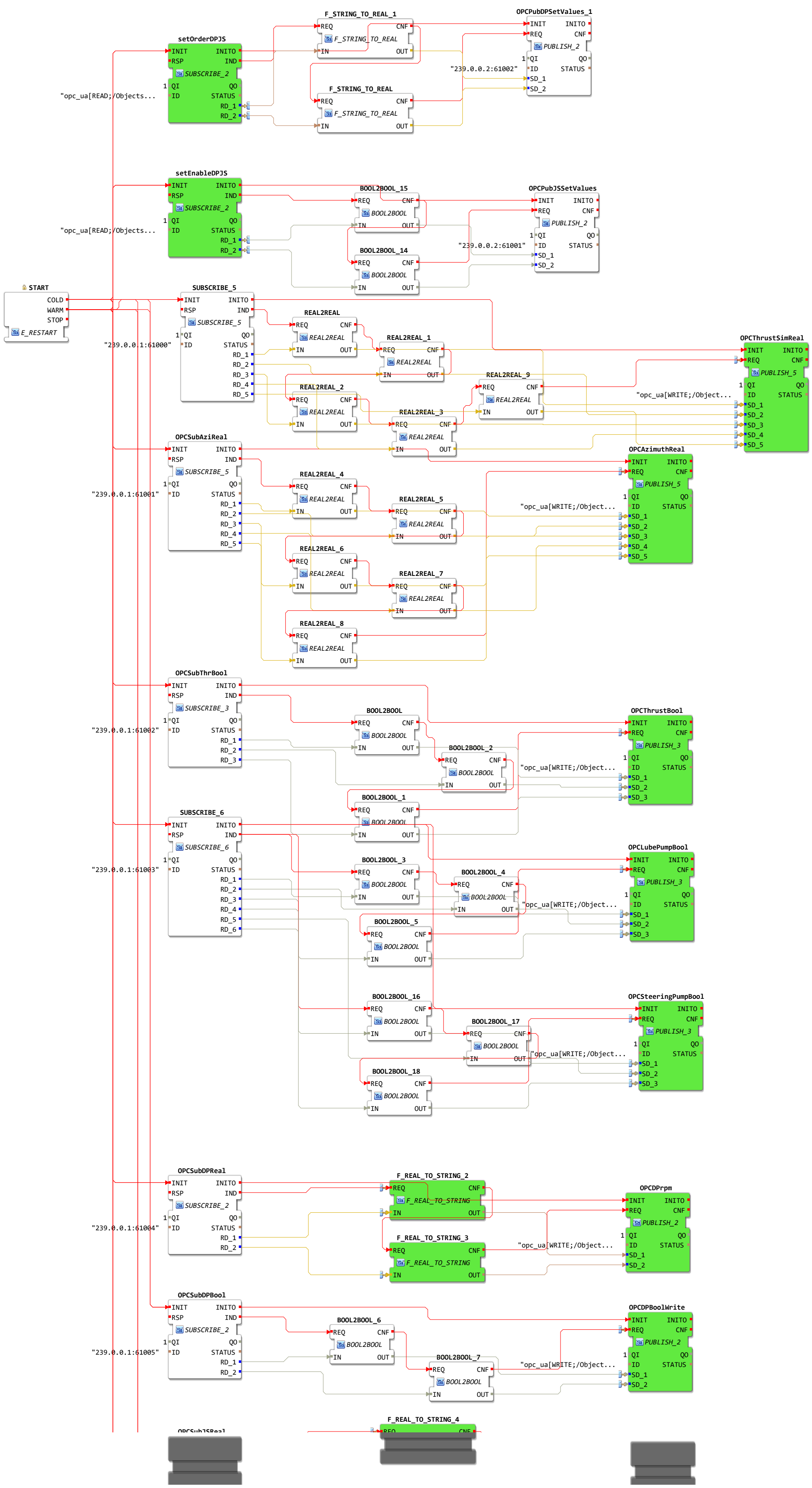
swingUpSimulator A200

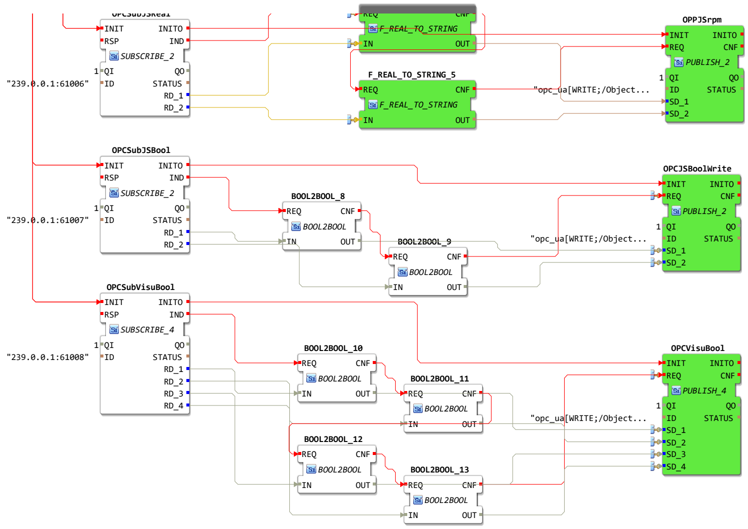





JSPitchFeedbackIn
JSPitchFeedbackInRemap
JSAziFeedbackIn
JSAziFeedbackInRemap
JSAziOrderRemap
JSAziOrder
OPCPubJSReal
CommonAlarm
FwdControl
AftControl
thrustRun
Visualisation
Visualisation_1
OPCPubVisuBool
"239.0.0.1:61006"
"239.0.0.1:61008"

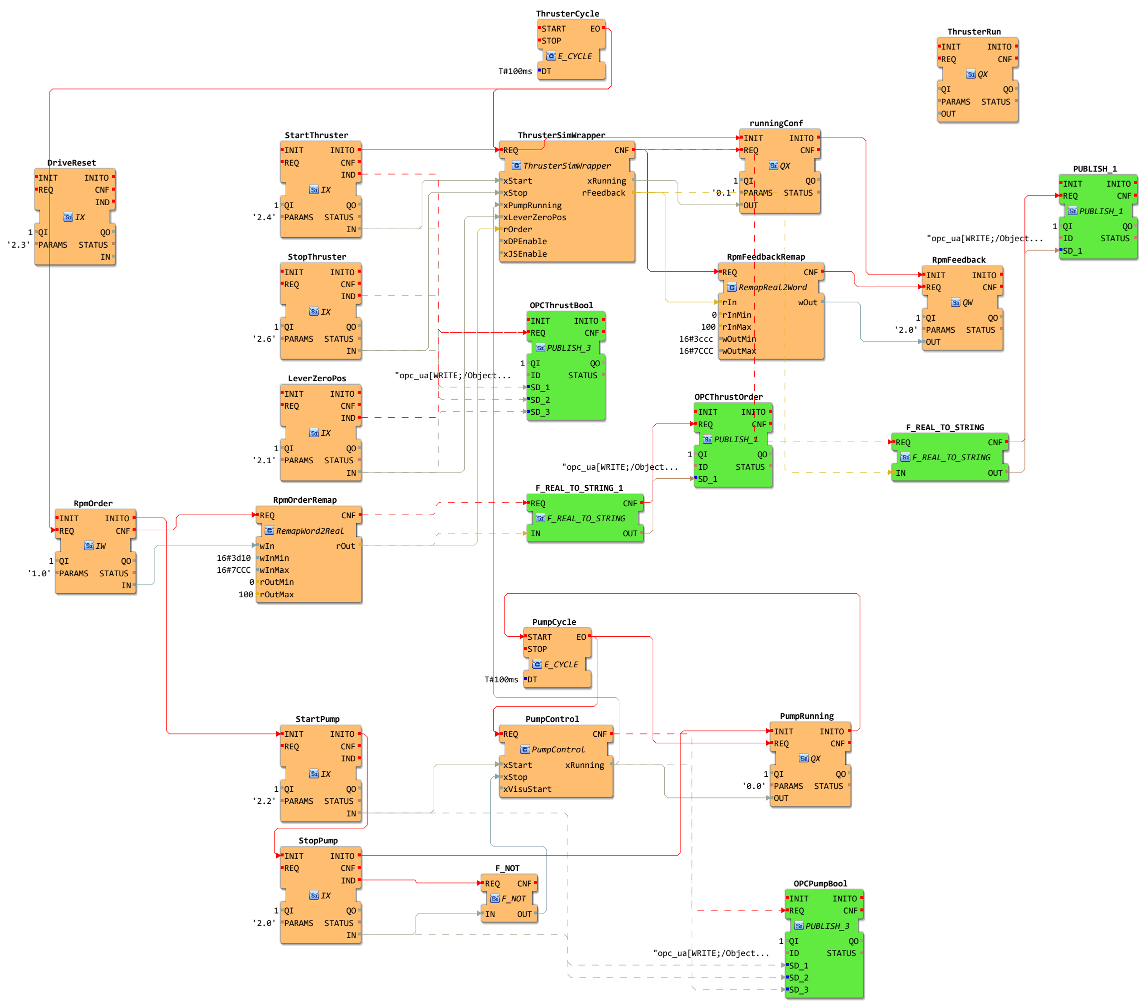
swingUpSimulator A200_OPC



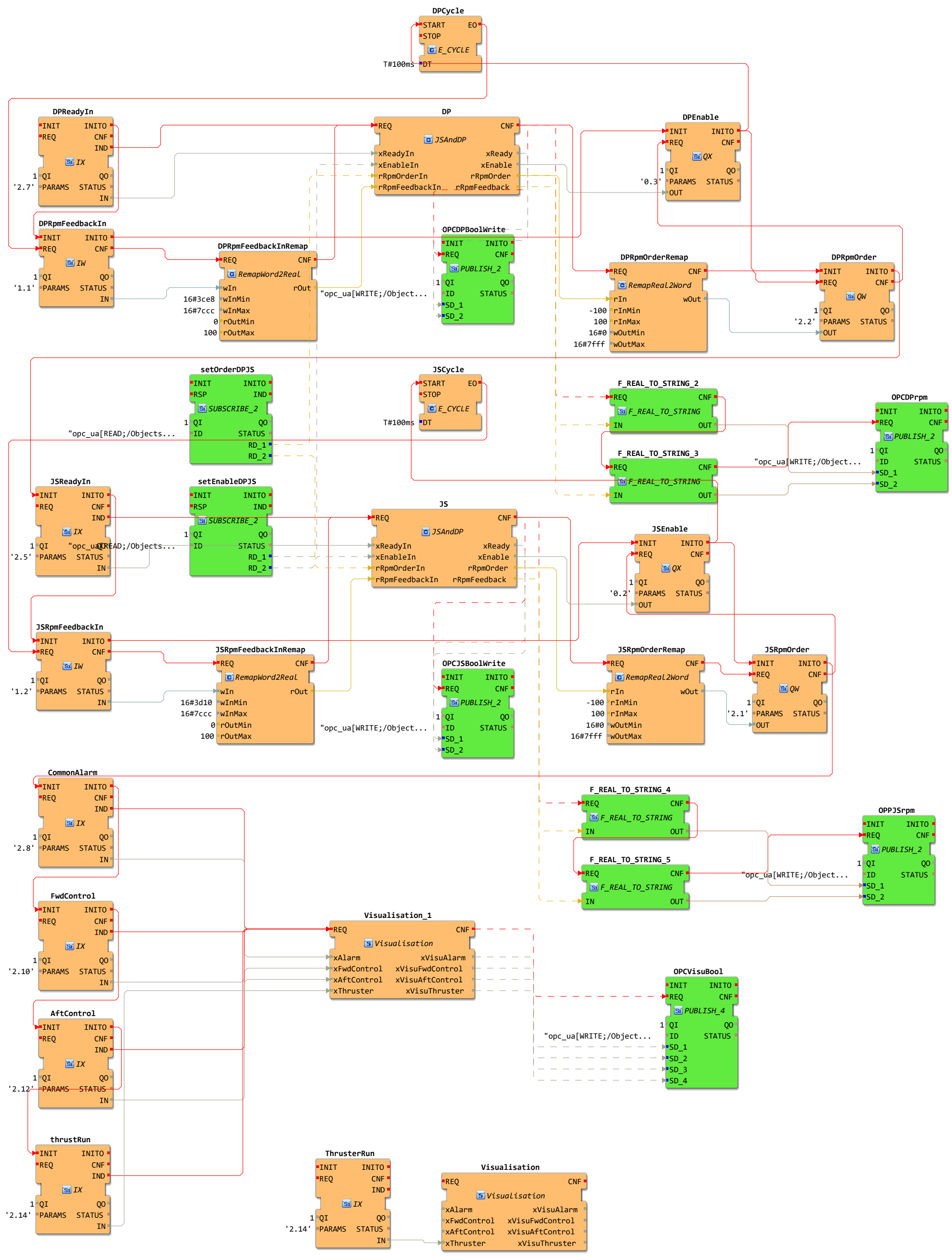


OPCSubJSReal
SUBSCRIBE_2
"239.0.0.1:61006"
F_REAL_TO_STRING
F_REAL_TO_STRING_5
OPPJSrpm
PUBLISH_2
"opc_ua[WRITE;/Object...
OPCSubJSBool
SUBSCRIBE_2
"239.0.0.1:61007"
BOOL2BOOL_8
BOOL2BOOL_9
OPCJSBoolWrite
PUBLISH_2
"opc_ua[WRITE;/Object...
OPCSubVisuBool
SUBSCRIBE_4
"239.0.0.1:61008"
BOOL2BOOL_10
BOOL2BOOL_11
BOOL2BOOL_12
BOOL2BOOL_13
OPCVisuBool
PUBLISH_4
"opc_ua[WRITE;/Object...

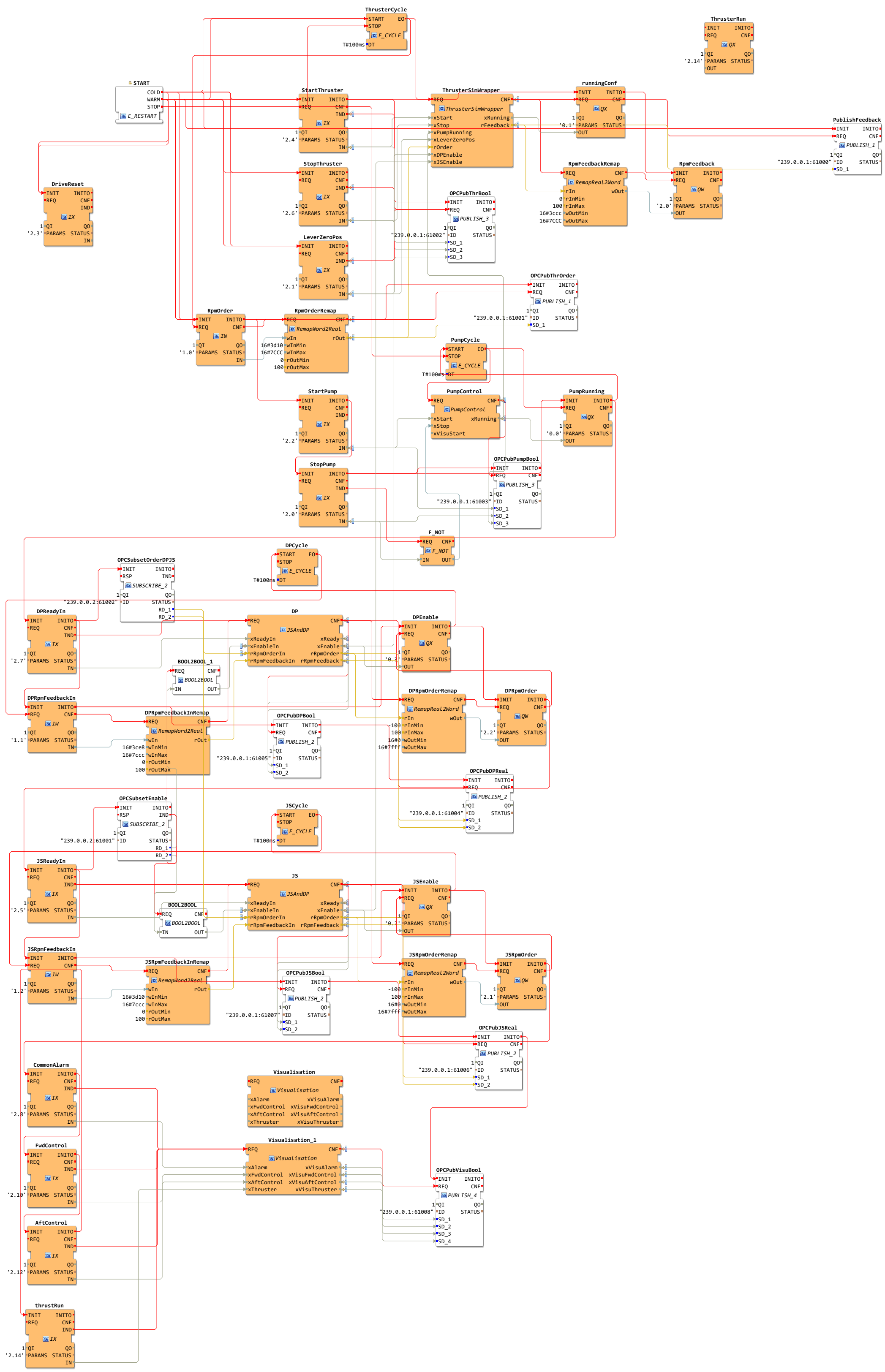
thrusterSimulator thrusterSim



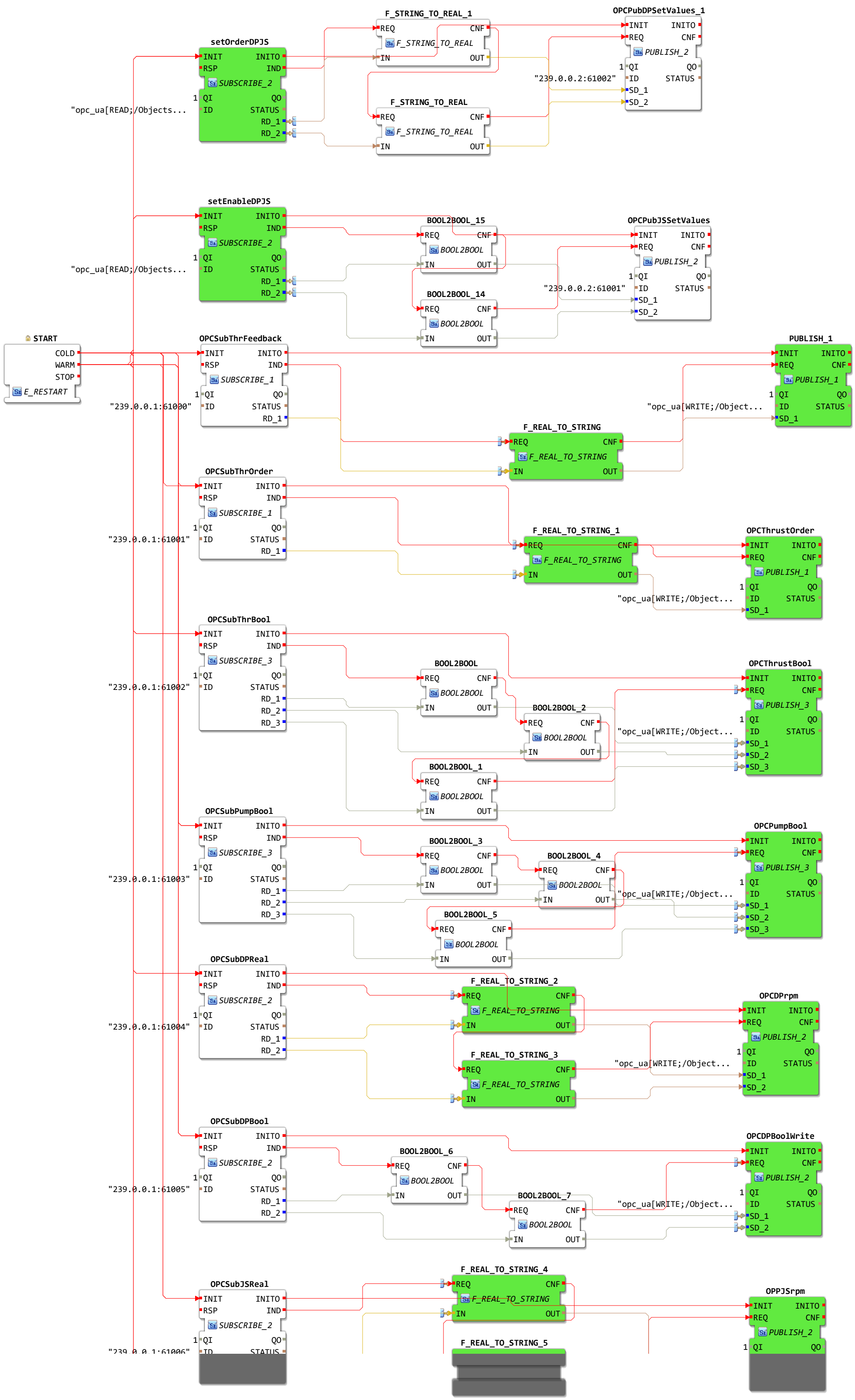
thrusterSimulator dpJsVisuSim

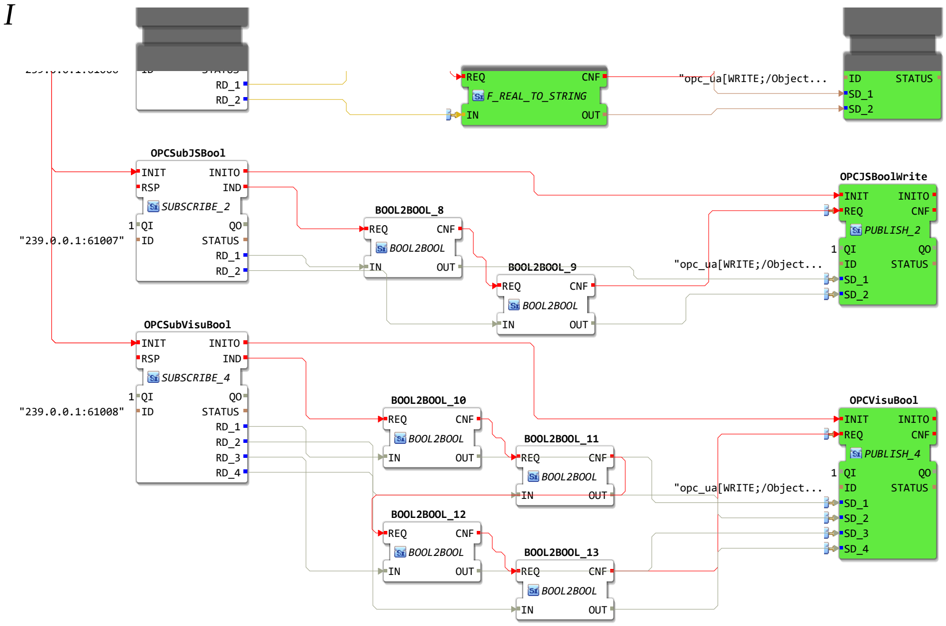


thrusterSimulator A200

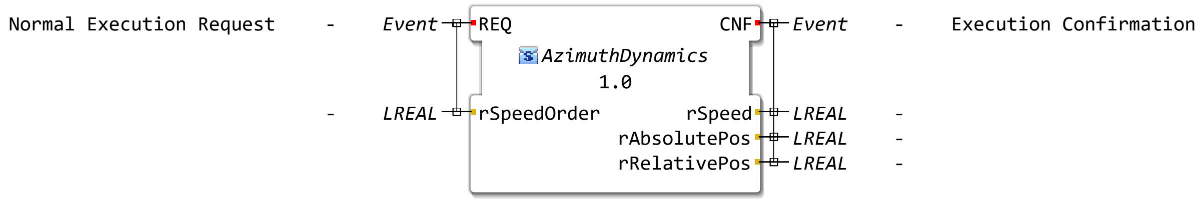


thrusterSimulator A200_OPC





REQ CNF
F_REAL_TO_STRING
IN OUT
"opc_ua[WRITE;/Object...
ID STATUS
SD_1
SD_2
RD_1
RD_2
OPCSubJSBool
INIT INITO
RSP IND
SUBSCRIBE_2
QI QO
ID STATUS
RD_1
RD_2
"239.0.0.1:61007"
BOOL2BOOL_8
BOOL2BOOL_9
BOOL2BOOL
OPCJSBoolWrite
PUBLISH_2
OPCSubVisuBool
SUBSCRIBE_4
RD_3
RD_4
"239.0.0.1:61008"
BOOL2BOOL_10
BOOL2BOOL_11
BOOL2BOOL_12
BOOL2BOOL_13
OPCVisuBool
PUBLISH_4
SD_3
SD_4


Normal Execution Request
-
Event
REQ
CNF
Event
-
Execution Confirmation
AzimuthDynamics
1.0
-
LREAL
rSpeedOrder
rSpeed
LREAL
-
rAbsolutePos
LREAL
-
rRelativePos
LREAL
-

Inputs

Name	Type	Comment	Initial Value	Array Size
rSpeedOrder	LREAL			

Outputs

Name	Type	Comment	Initial Value	Array Size
rSpeed	LREAL			
rAbsolutePos	LREAL			
rRelativePos	LREAL			

Internal Vars

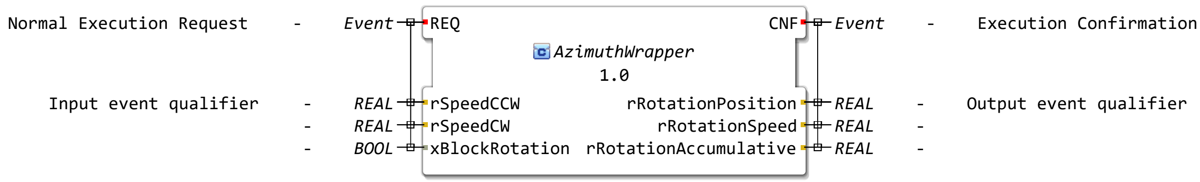
Internal FBs

Name	Type	Comment	Initial Value	Array Size
Integrator_DSTATE	LREAL		0	
Integrator_DSTATE_e	LREAL		0	
Filter_DSTATE	LREAL		0	
TransferFcn_states	LREAL		0	
Delay_DSTATE	LREAL		0	
rEQ0	BOOL			
q	LREAL			
rtb_error	LREAL			
rtb_Sum	LREAL			
temp1	LREAL			
u0	LREAL			
u1	LREAL			
u	LREAL			
tmp	LREAL			
rt_fmodd	LREAL			
rt_ceild	LREAL			
rt_ceild_u	LREAL			
rt_floord	LREAL			
rt_floord_u	LREAL			

```
(*
 *
 * File: azimuth_model.exp
 *
 * IEC 61131-3 Structured Text (ST) code generated for subsystem "azimuth_model/Subsystem/Subsystem discrete version 2"
 *
 * Model name                       : azimuth_model
 * Model version                    : 1.1
 * Model creator                    : espen
 * Model last modified by           : espen
 * Model last modified on           : Thu Apr 28 21:26:27 2022
 * Model sample time                : 0.1s
 * Subsystem name                   : azimuth_model/Subsystem/Subsystem discrete version 2
 * Subsystem sample time            : 0.1s
 * Simulink PLC Coder version       : 3.5 (R2021b) 14-May-2021
 * ST code generated on             : Fri Apr 29 13:08:41 2022
 *
 * Target IDE selection             : 3S CoDeSys 2.3
 * Test Bench included              : No
 *
 *)


(* Outputs for Atomic SubSystem: '<Root>/Subsystem discrete version 2' *)
(* Sum: '<S2>/Sum' incorporates:
 *  Delay: '<S2>/Delay' *)
rtb_error := rSpeedOrder - Delay_DSTATE;
(* Outport: '<Root>/Angle//rad' incorporates:
 *  DiscreteIntegrator: '<S2>/Integrator' *)
rAbsolutePos := Integrator_DSTATE;
(* MATLAB Function: '<S2>/Mod' incorporates:
 *  DiscreteIntegrator: '<S2>/Integrator' *)
(* MATLAB Function 'Subsystem discrete version 2/Subsystem/Mod': '<S3>:1' *)
(* '<S3>:1:3' y = mod(u, 2*pi); *)
IF Integrator_DSTATE = 0.0 THEN
    rRelativePos := 0.0;
ELSE

    // Set inputs of function
    u0 := Integrator_DSTATE;
    u1 := 6.2831853071795862;
    // START Inline implementation of rt_fmodd FUNCTION
    u := u0 / u1;

    // Set inputs for function
    rt_floord_u := q + u1;
    // START Inline implementation of rt_floord FUNCTION
    rt_floord := DINT_TO_LREAL(TRUNC(u));
    IF u = rt_floord THEN
        rt_floord := u;
    ELSIF u < 0.0 THEN
        rt_floord := rt_floord - 1.0;
    END_IF;
    // END rt_floord FUNCTION

    IF u1 <= rt_floord THEN
        rt_fmodd := u0 - (DINT_TO_LREAL(TRUNC(u)) * u1);
    ELSE
        tmp := ABS(u);
        IF tmp < 4.503599627370496E+15 THEN
            IF u >= 0.5 THEN

                // Set inputs for function
                rt_floord_u := u + 0.5;
                // START Inline implementation of rt_floord FUNCTION
                rt_floord := DINT_TO_LREAL(TRUNC(u));

                IF u = rt_floord THEN
                    rt_floord := u;
                ELSIF u < 0.0 THEN
                    rt_floord := rt_floord - 1.0;
```

```
                END_IF;
                // END rt_floord FUNCTION
                rt_fmodd := rt_floord;
                // ^^ Set outputs of function

            ELSIF u > -0.5 THEN
                rt_fmodd := 0.0;
            ELSE
                //rt_fmodd := rt_ceild(u := u - 0.5);

                // Set inputs for function
                rt_ceild_u := u - 0.5;
                // START Inline implementation of rt_ceild FUNCTION
                rt_ceild := DINT_TO_LREAL(TRUNC(rt_ceild_u));
                IF u = rt_ceild THEN
                    rt_ceild := rt_ceild_u;
                ELSIF u >= 0.0 THEN
                    rt_ceild := rt_ceild + 1.0;
                END_IF;
                 // END rt_ceild FUNCTION
                rt_fmodd := rt_ceild;
                // ^^ Set output of function

            END_IF;
        ELSE
            rt_fmodd := u;
        END_IF;
        IF ABS(u - rt_fmodd) <= (2.2204460492503131E-16 * tmp) THEN
            rt_fmodd := 0.0;
        ELSE
            rt_fmodd := (u - DINT_TO_LREAL(TRUNC(u))) * u1;
        END_IF;
    END_IF;
     // END rt_fmodd FUNCTION
    rRelativePos := rt_fmodd;
    // ^^ Set output of function

    rEQ0 := rRelativePos = 0.0;
    IF  NOT rEQ0 THEN
        q := ABS(Integrator_DSTATE / 6.2831853071795862);

        // Set inputs for function
        rt_floord_u := q + 0.5;
        // START Inline implementation of rt_floord FUNCTION
        rt_floord := DINT_TO_LREAL(TRUNC(u));

        IF u = rt_floord THEN
            rt_floord := u;
        ELSIF u < 0.0 THEN
            rt_floord := rt_floord - 1.0;
        END_IF;
        // END rt_floord FUNCTION
        temp1 := rt_floord;
        // ^^ Set outputs of function

        rEQ0 := ABS(q - temp1) <= (2.2204460492503131E-16 * q);
    END_IF;
    IF rEQ0 THEN
        rRelativePos := 0.0;
    ELSIF Integrator_DSTATE < 0.0 THEN
        rRelativePos := rRelativePos + 6.2831853071795862;
    END_IF;
END_IF;
(* End of MATLAB Function: '<S2>/Mod' *)

(* Gain: '<S39>/Filter Coefficient' incorporates:
 *  DiscreteIntegrator: '<S31>/Filter'
 *  Gain: '<S30>/Derivative Gain'
 *  Sum: '<S31>/SumD' *)
q := ((-1.14002691325679 * rtb_error) - Filter_DSTATE) * 0.558804748276024;
(* Sum: '<S45>/Sum' incorporates:
 *  DiscreteIntegrator: '<S36>/Integrator'
 *  Gain: '<S41>/Proportional Gain' *)
```

```
rtb_Sum := ((5.05380939935424 * rtb_error) + Integrator_DSTATE_e) + q;
(* Saturate: '<S43>/Saturation' *)
IF rtb_Sum > 250.0 THEN
    rtb_Sum := 250.0;
ELSIF rtb_Sum < -250.0 THEN
    rtb_Sum := -250.0;
END_IF;
(* End of Saturate: '<S43>/Saturation' *)

(* DiscreteTransferFcn: '<S2>/Transfer Fcn' *)
rtb_Sum := rtb_Sum - (-0.999956463532767 * TransferFcn_states);
rSpeed := (2.7210292021E-5 * rtb_Sum) + (2.7210292021E-5 * TransferFcn_states);
(* Update for Delay: '<S2>/Delay' incorporates:
 *  Gain: '<S2>/convert_to_percent'
 *  Gain: '<S2>/convert_to_rpm' *)
Delay_DSTATE := (9.5492965855137211 * rSpeed) * 25.0;
(* Update for DiscreteIntegrator: '<S2>/Integrator' *)
Integrator_DSTATE := (0.1 * rSpeed) + Integrator_DSTATE;
(* Update for DiscreteIntegrator: '<S36>/Integrator' incorporates:
 *  Gain: '<S33>/Integral Gain' *)
Integrator_DSTATE_e := ((0.025450831659634 * rtb_error) * 0.1) + Integrator_DSTATE_e;
(* Update for DiscreteIntegrator: '<S31>/Filter' *)
Filter_DSTATE := (0.1 * q) + Filter_DSTATE;
(* Update for DiscreteTransferFcn: '<S2>/Transfer Fcn' *)
TransferFcn_states := rtb_Sum;
(* End of Outputs for SubSystem: '<Root>/Subsystem discrete version 2' *)
```

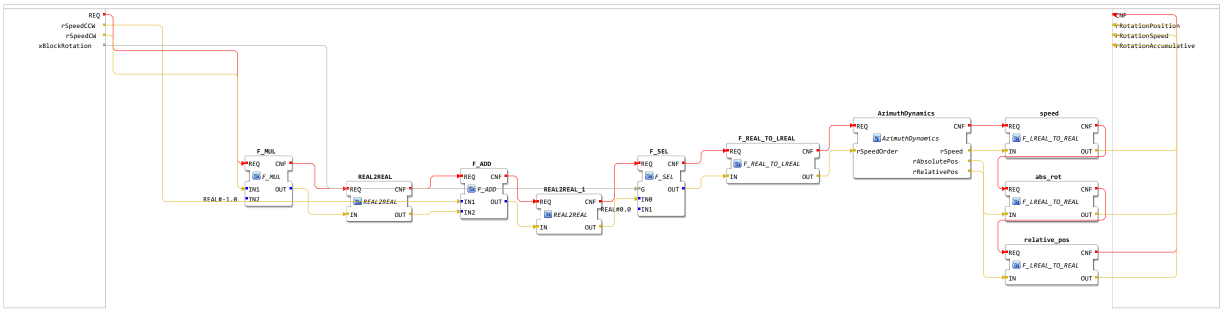

Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
AzimuthWrapper
1.0
Input event qualifier - REAL
rSpeedCCW
rRotationPosition
REAL - Output event qualifier
- REAL
rSpeedCW
rRotationSpeed
REAL -
- BOOL
xBlockRotation
rRotationAccumulative
REAL -

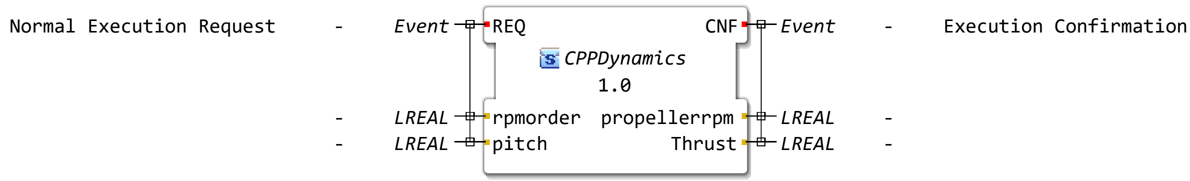
Inputs

Name	Type	Comment	Initial Value	Array Size
rSpeedCCW	REAL	Input event...		
rSpeedCW	REAL			
xBlockRotation	BOOL			

Outputs

Name	Type	Comment	Initial Value	Array Size
rRotationPosition	REAL	Output eve...		
rRotationSpeed	REAL			
rRotationAccumulative	REAL			


REQ
rSpeedCCW
rSpeedCW
xBlockRotation
F_MUL
REQ CNF
F_MUL
IN1 OUT
IN2
REAL#-1.0
REAL2REAL
REQ CNF
REAL2REAL
IN OUT
F_ADD
REQ CNF
F_ADD
IN1 OUT
IN2
REAL2REAL_1
REQ CNF
REAL2REAL
IN OUT
REAL#0.0
F_SEL
REQ CNF
F_SEL
G OUT
IN0
IN1
F_REAL_TO_LREAL
REQ CNF
F_REAL_TO_LREAL
IN OUT
AzimuthDynamics
REQ CNF
AzimuthDynamics
rSpeedOrder rSpeed
rAbsolutePos
rRelativePos
speed
REQ CNF
F_LREAL_TO_REAL
IN OUT
abs_rot
REQ CNF
F_LREAL_TO_REAL
IN OUT
relative_pos
REQ CNF
F_LREAL_TO_REAL
IN OUT
CNF
rRotationPosition
rRotationSpeed
rRotationAccumulative


Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
CPPDynamics
1.0
- LREAL
rpmorder
propellerrpm
LREAL -
- LREAL
pitch
Thrust
LREAL -

Inputs

Name	Type	Comment	Initial Value	Array Size
rpmorder	LREAL			
pitch	LREAL			

Outputs

Name	Type	Comment	Initial Value	Array Size
propellerrpm	LREAL			
Thrust	LREAL			

Internal Vars

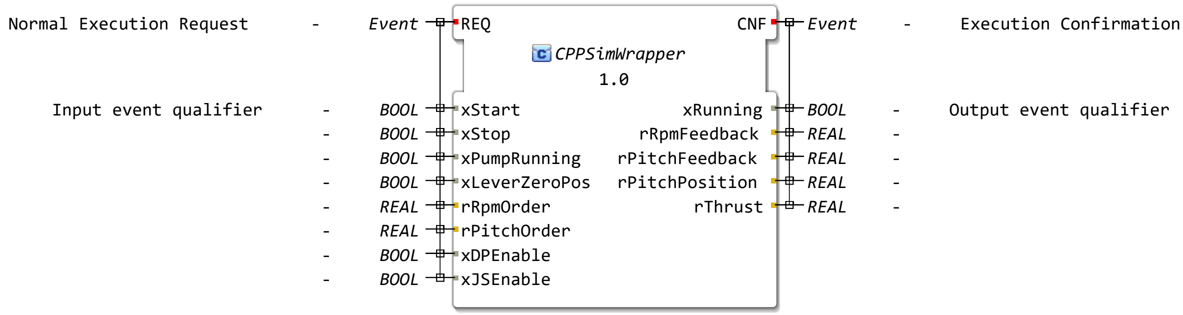
Internal FBs

Name	Type	Comment	Initial Value	Array Size
Integrator_DSTATE	LREAL		0.0	
Integrator_DSTATE_I	LREAL		0.0	
Filter_DSTATE	LREAL		0.0	
Integrator1_DSTATE	LREAL		0.0	
U_p	LREAL			
theta	LREAL			
alpha_e	LREAL			
Lift	LREAL			
Drag	LREAL			
rtb_Gain1	LREAL			
rtb_Sum5	LREAL			
rtb_Saturation	LREAL			
Integrator_DSTATE_tmp	LREAL			
Lift_tmp	LREAL			
rtb_T_tmp	LREAL			
rtb_FilterCoefficient	LREAL			
rt_atan2d	LREAL			
u0	LREAL			
u1	LREAL			

```
(*
 *
 * File: CPP_Healey_TT_2000_Simulink_v2.exp
 *
 * IEC 61131-3 Structured Text (ST) code generated for subsystem "CPP_Healey_TT_2000_Simulink_v2/Subsystem1/Subsystem1 discrete version 2"
 *
 * Model name                       : CPP_Healey_TT_2000_Simulink_v2
 * Model version                    : 1.37
 * Model creator                    : haram
 * Model last modified by           : espen
 * Model last modified on           : Fri May 06 09:22:20 2022
 * Model sample time                : 0.01s
 * Subsystem name                   : CPP_Healey_TT_2000_Simulink_v2/Subsystem1/Subsystem1 discrete version 2
 * Subsystem sample time            : 0.01s
 * Simulink PLC Coder version       : 3.5 (R2021b) 14-May-2021
 * ST code generated on             : Fri May 06 09:26:58 2022
 *
 * Target IDE selection             : 3S CoDeSys 2.3
 * Test Bench included              : No
 *
 *)


        (* Outputs for Atomic SubSystem: '<Root>/Subsystem1 discrete version 2' *)
        (* Outputs for Atomic SubSystem: '<S1>/Subsystem1' *)
        (* Gain: '<S2>/Gain1' incorporates:
         *  DiscreteIntegrator: '<S2>/Integrator' *)
        (* MATLAB Function 'Subsystem1 discrete version 2/Subsystem1/MATLAB Function2': '<S5>:1' *)
        (* '<S5>:1:2' inMin = -100; *)
        (* '<S5>:1:3' inMax = 100; *)
        (* '<S5>:1:4' outMin = -23; *)
        (* '<S5>:1:5' outMax = 23; *)
        (* '<S5>:1:7' w_order = (((pcg_order - inMin) * (outMax - outMin)) / (inMax - inMin)) + outMin; *)
        rtb_Gain1 := 0.16666666666666666 * Integrator_DSTATE;
        (* Sum: '<S2>/Sum5' incorporates:
         *  MATLAB Function: '<S2>/MATLAB Function2' *)
        rtb_Sum5 := ((((rpmorder - -100.0) * 46.0) / 200.0) + -23.0) - rtb_Gain1;
        (* Gain: '<S42>/Filter Coefficient' incorporates:
         *  DiscreteIntegrator: '<S34>/Filter'
         *  Gain: '<S33>/Derivative Gain'
         *  Sum: '<S34>/SumD' *)
        rtb_FilterCoefficient := ((400.0 * rtb_Sum5) - Filter_DSTATE) * 0.995024875621891;
        (* Sum: '<S48>/Sum' incorporates:
         *  DiscreteIntegrator: '<S39>/Integrator'
         *  Gain: '<S44>/Proportional Gain' *)
        rtb_Saturation := ((100.0 * rtb_Sum5) + Integrator_DSTATE_l) + rtb_FilterCoefficient;
        (* Saturate: '<S46>/Saturation' *)
        IF rtb_Saturation > 1500.0 THEN
            rtb_Saturation := 1500.0;
        ELSIF rtb_Saturation < -1500.0 THEN
            rtb_Saturation := -1500.0;
        END_IF;
        (* End of Saturate: '<S46>/Saturation' *)

        (* MATLAB Function: '<S2>/MATLAB Function' incorporates:
         *  Constant: '<S2>/Constant4'
         *  Constant: '<S2>/Constant6'
         *  DiscreteIntegrator: '<S2>/Integrator1' *)
        (* rho = 998;            % water density *)
        (* D = 0.0762;                 % propeller diameter in meter *)
        (* MATLAB Function 'Subsystem1 discrete version 2/Subsystem1/MATLAB Function': '<S3>:1' *)
        (* '<S3>:1:5' R = D/2; *)
        (* propeller radius in meter *)
        (* A = 3.14 * R^2;        % tunnel cross-sectional area *)
        (* '<S3>:1:7' U_p = 0.7 * R * w_p; *)
        U_p := 0.7 * rtb_Gain1;
```

```
          (* propeller velocity *)
          (* '<S3>:1:8' V_sqr = U_a^2 + U_p^2; *)
          (* total relative velocity squared *)
          (* '<S3>:1:9' C_lmax = 1.75; *)

          (* '<S3>:1:10' C_dmax = 1.2; *)

          (* alpha_e = ((pi / 2) - pitch) - atan2(U_a,U_p); % effective angle of attack *)
          (* theta = pitch - alpha_e; *)
          (* '<S3>:1:14' theta = atan2(U_a,U_p); *)

           //theta := rt_atan2d(u0 := Integrator1_DSTATE, u1 := U_p);

    // Set inputs for function
  u0 := Integrator1_DSTATE;
  u1 := U_p;
  // START Inline implementation of rt_atan2d FUNCTION
  IF u1 > 0.0 THEN
      rt_atan2d := ATAN(u0 / u1);
  ELSIF u1 < 0.0 THEN
      IF u0 >= 0.0 THEN
          rt_atan2d := 1.0;
      ELSE
          rt_atan2d := -1.0;
      END_IF;
      rt_atan2d := ATAN(u0 / u1) + (rt_atan2d * 3.1415926535897931);
  ELSE
      IF u0 < 0.0 THEN
          rt_atan2d := -1.0;
      ELSIF u0 > 0.0 THEN
          rt_atan2d := 1.0;
      ELSE
          rt_atan2d := u0;
      END_IF;
      rt_atan2d := rt_atan2d * 1.5707963267948966;
  END_IF;
  // END rt_floord FUNCTION
  theta := rt_atan2d;
  // Set outputs for function




          (* '<S3>:1:15' alpha_e = ((pi / 2) - pitch) - theta; *)
          alpha_e := (1.5707963267948966 - pitch) - theta;
          (* effective angle of attack *)
          (* '<S3>:1:17' Lift = 0.5 * rho * V_sqr * A * C_lmax * sin(2 * alpha_e); *)
          Lift_tmp := (512.5 * ((Integrator1_DSTATE * Integrator1_DSTATE) + (U_p * U_p))) * 3.14;
          Lift := (Lift_tmp * 1.75) * SIN(2.0 * alpha_e);
          (* '<S3>:1:18' Drag = 0.5 * rho * V_sqr * A * C_dmax * (1 - cos(2 * alpha_e)); *)
          Drag := (Lift_tmp * 1.2) * (1.0 - COS(2.0 * alpha_e));
          (* '<S3>:1:20' Q = 0.7 * R * ((Lift * sin(theta)) + (Drag * cos(theta))); *)
          (* '<S3>:1:21' T = (Lift * cos(theta)) - (Drag * sin(theta)); *)
          Lift_tmp := SIN(theta);
          rtb_T_tmp := COS(theta);
          Thrust := (Lift * rtb_T_tmp) - (Drag * Lift_tmp);
          (* Update for DiscreteIntegrator: '<S2>/Integrator' incorporates:
           *  Gain: '<S2>/Gain'
           *  Gain: '<S2>/Gain2'
           *  Gain: '<S2>/Gain3'
           *  MATLAB Function: '<S2>/MATLAB Function'
           *  Sum: '<S2>/Sum'
           *  Sum: '<S2>/Sum1' *)
          (* MATLAB Function 'Subsystem1 discrete version 2/Subsystem1/MATLAB Function1': '<S4>:1' *)
          (* '<S4>:1:3' x = K4*U_a_diff*abs(U_a_diff); *)
          (* MATLAB Function 'Subsystem1 discrete version 2/Subsystem1/MATLAB Function3': '<S6>:1' *)
          (* '<S6>:1:2' inMin = -23; *)
          (* '<S6>:1:3' inMax = 23; *)
```

```
        (* '<S6>:1:4' outMin = -100; *)
        (* '<S6>:1:5' outMax = 100; *)
        (* '<S6>:1:7' pcg_feedback = (((w_feedback - inMin) * (outMax - outMin)) /
        (inMax - inMin)) + outMin; *)
        Integrator_DSTATE := ((((0.068994657130412557 * rtb_Saturation) - ((((Lift *
        Lift_tmp) + (Drag * rtb_T_tmp)) * 0.7) * 0.00033155765787670478)) -
        (0.028748211460446964 * Integrator_DSTATE)) * 0.01) + Integrator_DSTATE;
        (* Update for DiscreteIntegrator: '<S39>/Integrator' incorporates:
         *  Gain: '<S36>/Integral Gain' *)
        Integrator_DSTATE_l := ((35.0 * rtb_Sum5) * 0.01) + Integrator_DSTATE_l;
        (* Update for DiscreteIntegrator: '<S34>/Filter' *)
        Filter_DSTATE := (0.01 * rtb_FilterCoefficient) + Filter_DSTATE;
        (* Update for DiscreteIntegrator: '<S2>/Integrator1' incorporates:
         *  Constant: '<S2>/Constant1'
         *  Gain: '<S2>/Gain4'
         *  MATLAB Function: '<S2>/MATLAB Function1'
         *  Sum: '<S2>/Sum2' *)
        Integrator1_DSTATE := (((Thrust - ((643.7 * Integrator1_DSTATE) *
        ABS(Integrator1_DSTATE))) * 0.00024856299518409195) * 0.01) +
        Integrator1_DSTATE;
        (* Outport: '<Root>/propeller rpm' incorporates:
         *  MATLAB Function: '<S2>/MATLAB Function3' *)
        propellerrpm := (((rtb_Gain1 - -23.0) * 200.0) / 46.0) + -100.0;
        (* End of Outputs for SubSystem: '<S1>/Subsystem1' *)
        (* End of Outputs for SubSystem: '<Root>/Subsystem1 discrete version 2' *)
```

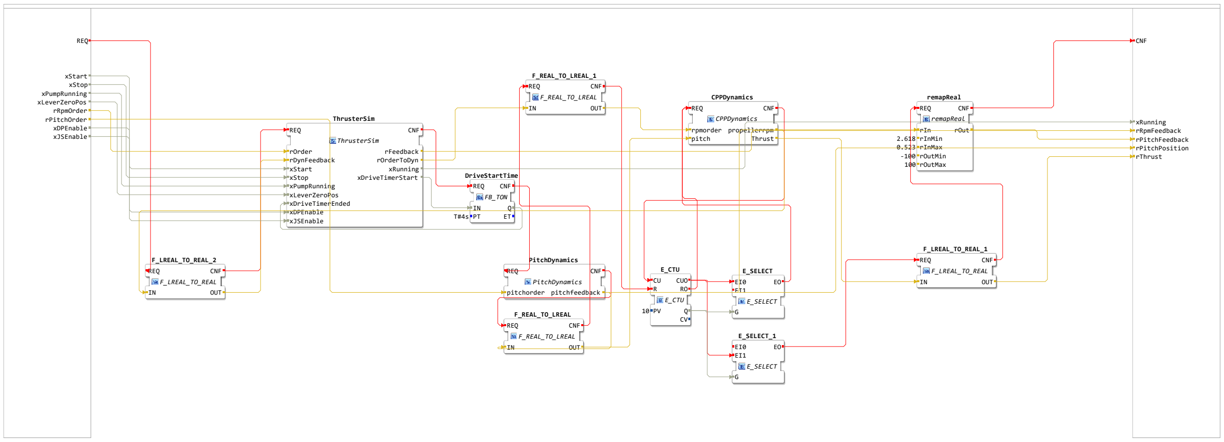

Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
CPPSimWrapper
1.0
Input event qualifier - BOOL
xStart
xRunning
BOOL - Output event qualifier
- BOOL
xStop
rRpmFeedback
REAL -
- BOOL
xPumpRunning
rPitchFeedback
REAL -
- BOOL
xLeverZeroPos
rPitchPosition
REAL -
- REAL
rRpmOrder
rThrust
REAL -
- REAL
rPitchOrder
- BOOL
xDPEnable
- BOOL
xJSEnable

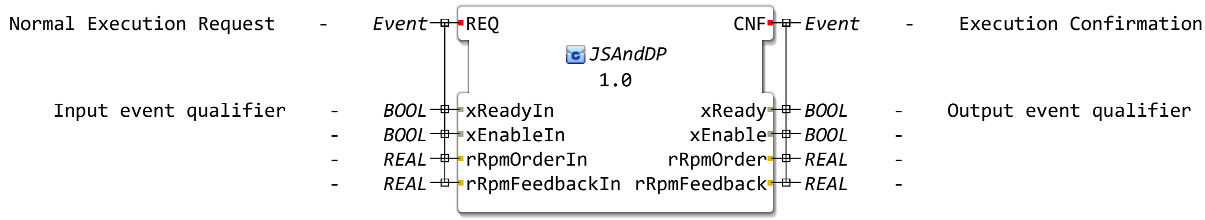
Inputs

Name	Type	Comment	Initial Value	Array Size
xStart	BOOL	Input event...	0	
xStop	BOOL		0	
xPumpRunning	BOOL		0	
xLeverZeroPos	BOOL		0	
rRpmOrder	REAL		0	
rPitchOrder	REAL			
xDPEnable	BOOL		0	
xJSEnable	BOOL		0	

Outputs

Name	Type	Comment	Initial Value	Array Size
xRunning	BOOL	Output eve...	0	
rRpmFeedback	REAL		0	
rPitchFeedback	REAL			
rPitchPosition	REAL			
rThrust	REAL			



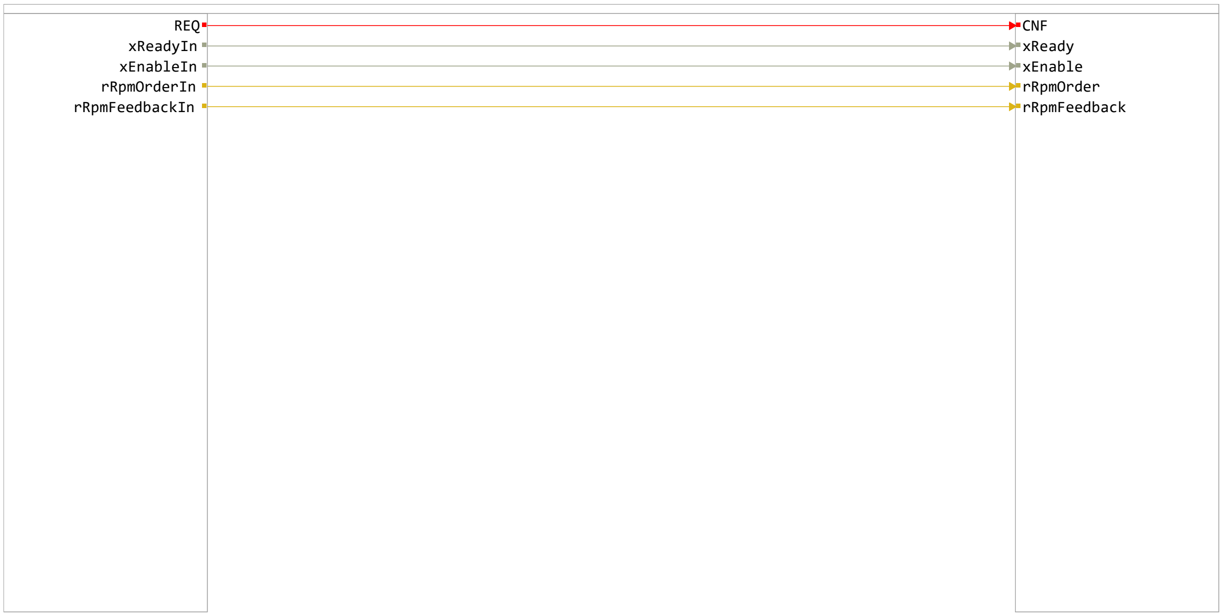

Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
JSAndDP
1.0
Input event qualifier - BOOL
xReadyIn
xReady
BOOL - Output event qualifier
- BOOL
xEnableIn
xEnable
BOOL -
- REAL
rRpmOrderIn
rRpmOrder
REAL -
- REAL
rRpmFeedbackIn
rRpmFeedback
REAL -

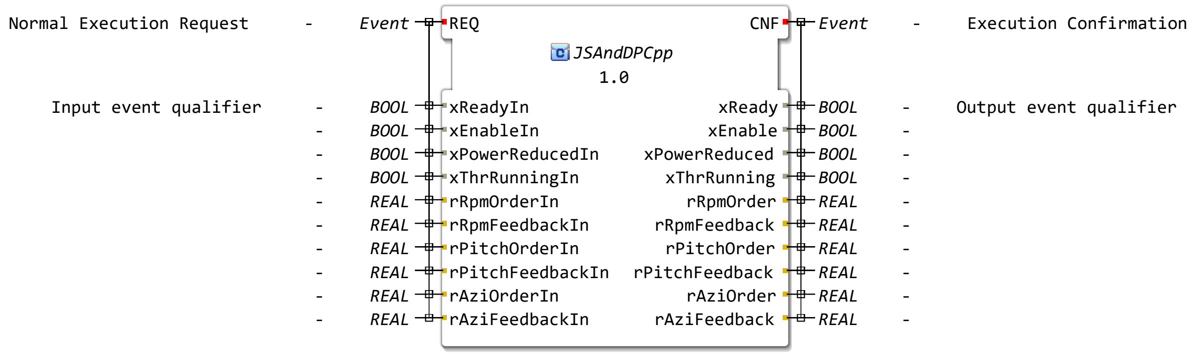
Inputs

Name	Type	Comment	Initial Value	Array Size
xReadyIn	BOOL	Input event...		
xEnableIn	BOOL			
rRpmOrderIn	REAL			
rRpmFeedbackIn	REAL			

Outputs

Name	Type	Comment	Initial Value	Array Size
xReady	BOOL	Output eve...		
xEnable	BOOL			
rRpmOrder	REAL			
rRpmFeedback	REAL			


REQ
xReadyIn
xEnableIn
rRpmOrderIn
rRpmFeedbackIn
CNF
xReady
xEnable
rRpmOrder
rRpmFeedback


Normal Execution Request - Event REQ
CNF Event - Execution Confirmation
JSAndDPCpp
1.0
Input event qualifier - BOOL xReadyIn
- BOOL xEnableIn
- BOOL xPowerReducedIn
- BOOL xThrRunningIn
- REAL rRpmOrderIn
- REAL rRpmFeedbackIn
- REAL rPitchOrderIn
- REAL rPitchFeedbackIn
- REAL rAziOrderIn
- REAL rAziFeedbackIn
xReady BOOL - Output event qualifier
xEnable BOOL -
xPowerReduced BOOL -
xThrRunning BOOL -
rRpmOrder REAL -
rRpmFeedback REAL -
rPitchOrder REAL -
rPitchFeedback REAL -
rAziOrder REAL -
rAziFeedback REAL -

Inputs

Name	Type	Comment	Initial Value	Array Size
xReadyIn	BOOL	Input event...		
xEnableIn	BOOL			
xPowerReducedIn	BOOL			
xThrRunningIn	BOOL			
rRpmOrderIn	REAL			
rRpmFeedbackIn	REAL			
rPitchOrderIn	REAL			
rPitchFeedbackIn	REAL			
rAziOrderIn	REAL			
rAziFeedbackIn	REAL			

Outputs

Name	Type	Comment	Initial Value	Array Size
xReady	BOOL	Output eve...		
xEnable	BOOL			
xPowerReduced	BOOL			
xThrRunning	BOOL			
rRpmOrder	REAL			
rRpmFeedback	REAL			
rPitchOrder	REAL			
rPitchFeedback	REAL			
rAziOrder	REAL			
rAziFeedback	REAL			


REQ
xReadyIn
xEnableIn
xPowerReducedIn
xThrRunningIn
rRpmOrderIn
rRpmFeedbackIn
rPitchOrderIn
rPitchFeedbackIn
rAziOrderIn
rAziFeedbackIn
CNF
xReady
xEnable
xPowerReduced
xThrRunning
rRpmOrder
rRpmFeedback
rPitchOrder
rPitchFeedback
rAziOrder
rAziFeedback


Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
PitchDynamics
1.0
- REAL
pitchorder
pitchfeedback
REAL -

Inputs

Name	Type	Comment	Initial Value	Array Size
pitchorder	REAL			

Outputs

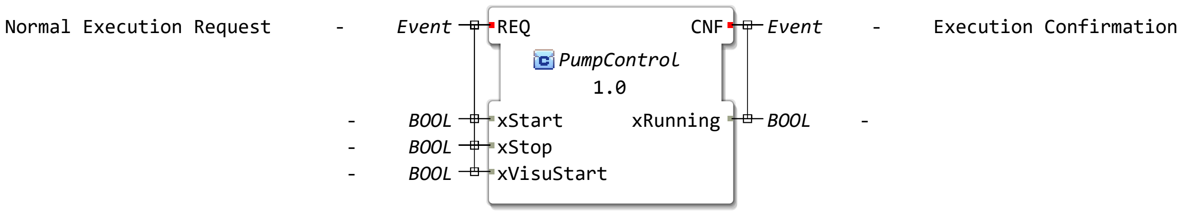
Name	Type	Comment	Initial Value	Array Size
pitchfeedback	REAL			

Internal Vars

Internal FBs

Name	Type	Comment	Initial Value	Array Size
Memory_PreviousInput	LREAL		0	
Integrator_DSTATE	LREAL		0	
Filter_DSTATE	LREAL		0	
PitchDynamics_states	LREAL		0	
rtb_Sum4	LREAL			
rtb_FilterCoefficient	LREAL			
rtb_PitchDynamics	LREAL			
PitchDynamics_tmp	LREAL			

```
(*
 *
 * File: CPP_Healey_TT_2000_Simulink_v2.exp
 *
 * IEC 61131-3 Structured Text (ST) code generated for subsystem "CPP_Healey_TT_2000_Simulink_v2/Subsystem/Subsystem discrete version 2"
 *
 * Model name                       : CPP_Healey_TT_2000_Simulink_v2
 * Model version                    : 1.38
 * Model creator                    : haram
 * Model last modified by           : espen
 * Model last modified on           : Fri May 06 18:23:50 2022
 * Model sample time                : 0.01s
 * Subsystem name                   : CPP_Healey_TT_2000_Simulink_v2/Subsystem/Subsystem discrete version 2
 * Subsystem sample time            : 0.1s
 * Simulink PLC Coder version       : 3.5 (R2021b) 14-May-2021
 * ST code generated on             : Fri May 06 19:07:01 2022
 *
 * Target IDE selection             : 3S CoDeSys 2.3
 * Test Bench included              : No
 *
 *)

(* Outputs for Atomic SubSystem: '<Root>/Subsystem discrete version 3' *)
(* Sum: '<S2>/Sum4' incorporates:
 *  Memory: '<S2>/Memory' *)
rtb_Sum4 := pitchorder - Memory_PreviousInput;
(* Gain: '<S38>/Filter Coefficient' incorporates:
 *  DiscreteIntegrator: '<S30>/Filter'
 *  Gain: '<S29>/Derivative Gain'
 *  Sum: '<S30>/SumD' *)
rtb_FilterCoefficient := ((0.013772211459469 * rtb_Sum4) - Filter_DSTATE) * 0.512025506073858;
(* DiscreteTransferFcn: '<S2>/Pitch Dynamics' incorporates:
 *  DiscreteIntegrator: '<S35>/Integrator'
 *  Gain: '<S40>/Proportional Gain'
 *  Sum: '<S44>/Sum' *)
PitchDynamics_tmp := (((0.025192867734574 * rtb_Sum4) + Integrator_DSTATE) + rtb_FilterCoefficient) - (-0.987577639751553 * PitchDynamics_states);
rtb_PitchDynamics := (0.009316770186335 * PitchDynamics_tmp) + (0.009316770186335 * PitchDynamics_states);
(* Sum: '<S2>/Add' incorporates:
 *  Constant: '<S2>/Minimum Pitch1'
 *  Gain: '<S2>/Gain' *)
pitchfeedback := ( -rtb_PitchDynamics) + 1.5707963267948966;
(* Saturate: '<S2>/Saturation' incorporates:
 *  Constant: '<S2>/Minimum Pitch1'
 *  Gain: '<S2>/Gain'
 *  Sum: '<S2>/Add' *)
IF (( -rtb_PitchDynamics) + 1.5707963267948966) > 2.6179938779914944 THEN
    (* Outport: '<Root>/pitch feedback' *)
    pitchfeedback := 2.6179938779914944;
ELSIF (( -rtb_PitchDynamics) + 1.5707963267948966) < 0.52359877559829882 THEN
    (* Outport: '<Root>/pitch feedback' *)
    pitchfeedback := 0.52359877559829882;
END_IF;
(* End of Saturate: '<S2>/Saturation' *)

(* Update for Memory: '<S2>/Memory' incorporates:
 *  DiscreteTransferFcn: '<S2>/Pitch Feedback Percentage' *)
Memory_PreviousInput := rtb_PitchDynamics * 95.492965855137214;
(* Update for DiscreteIntegrator: '<S35>/Integrator' incorporates:
 *  Gain: '<S32>/Integral Gain' *)
Integrator_DSTATE := ((0.004706440151022 * rtb_Sum4) * 0.1) + Integrator_DSTATE;
(* Update for DiscreteIntegrator: '<S30>/Filter' *)
Filter_DSTATE := (0.1 * rtb_FilterCoefficient) + Filter_DSTATE;
(* Update for DiscreteTransferFcn: '<S2>/Pitch Dynamics' *)
PitchDynamics_states := PitchDynamics_tmp;
(* End of Outputs for SubSystem: '<Root>/Subsystem discrete version 3' *)
```

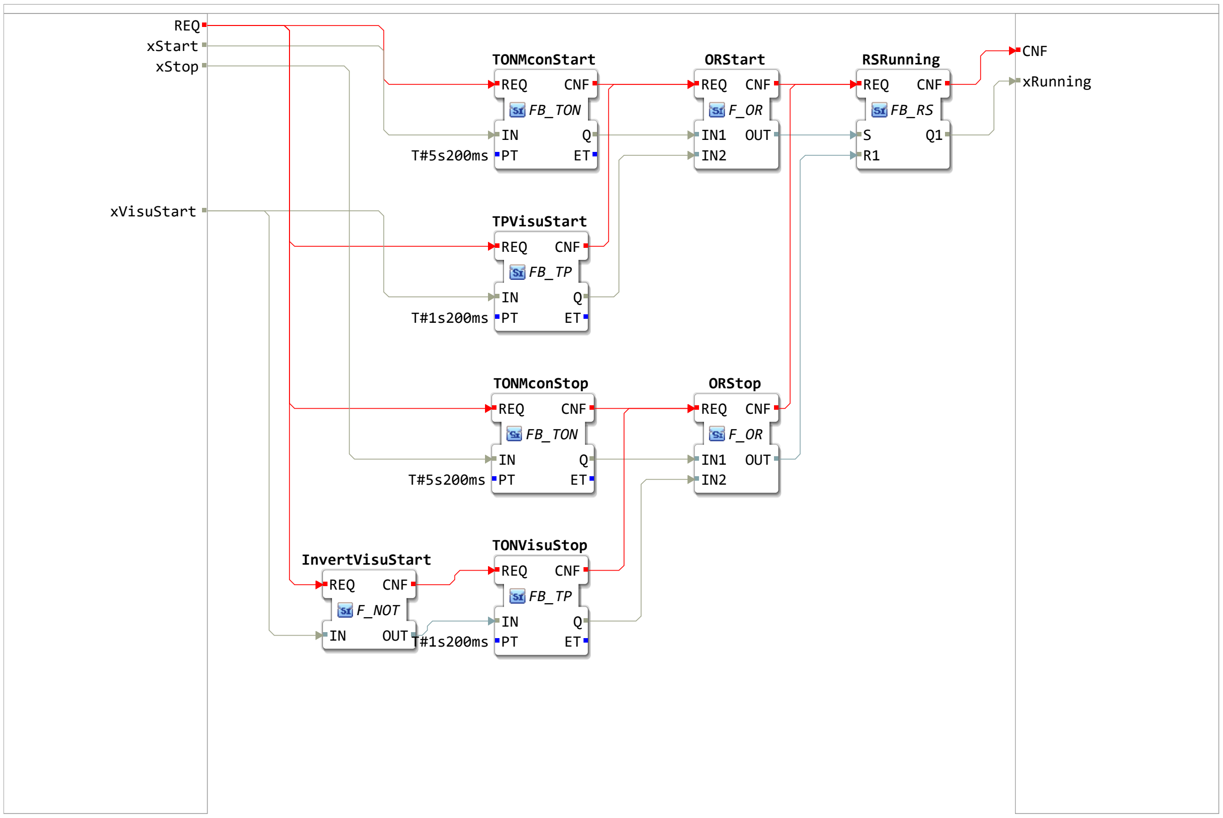

Normal Execution Request
-
Event
REQ
CNF
Event
-
Execution Confirmation
PumpControl
1.0
-
BOOL
xStart
xRunning
BOOL
-
-
BOOL
xStop
-
BOOL
xVisuStart

Inputs

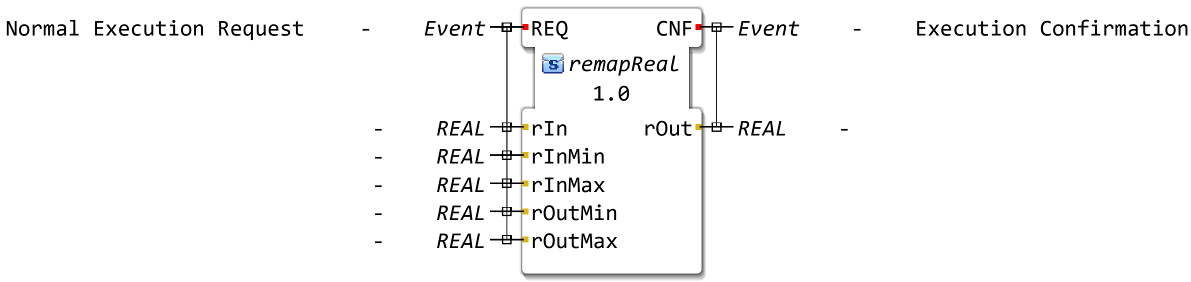
Name	Type	Comment	Initial Value	Array Size
xStart	BOOL			
xStop	BOOL			
xVisuStart	BOOL			

Outputs

Name	Type	Comment	Initial Value	Array Size
xRunning	BOOL			



REQ
xStart
xStop
xVisuStart
TONMconStart
REQ CNF
FB_TON
IN Q
T#5s200ms PT ET
ORStart
REQ CNF
F_OR
IN1 OUT
IN2
RSRunning
REQ CNF
FB_RS
S Q1
R1
CNF
xRunning
TPVisuStart
REQ CNF
FB_TP
IN Q
T#1s200ms PT ET
TONMconStop
REQ CNF
FB_TON
IN Q
T#5s200ms PT ET
ORStop
REQ CNF
F_OR
IN1 OUT
IN2
InvertVisuStart
REQ CNF
F_NOT
IN OUT
TONVisuStop
REQ CNF
FB_TP
IN Q
T#1s200ms PT ET


Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
remapReal
1.0
- REAL
rIn
rOut
REAL -
- REAL
rInMin
- REAL
rInMax
- REAL
rOutMin
- REAL
rOutMax

Inputs

Name	Type	Comment	Initial Value	Array Size
rIn	REAL			
rInMin	REAL			
rInMax	REAL			
rOutMin	REAL			
rOutMax	REAL			

Outputs

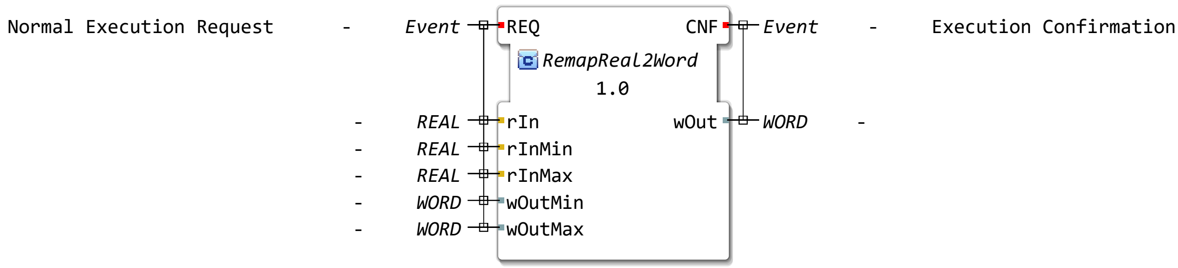
Name	Type	Comment	Initial Value	Array Size
rOut	REAL			

Internal Vars

Internal FBs

Name	Type	Comment	Initial Value	Array Size
rA	REAL		0	
rB	REAL		0	

```
// Calculate the step length
IF (rInMax - rInMin) <> 0 THEN
    rA := (rOutMax - rOutMin) / (rInMax - rInMin);

    // Calculate the inital step
    rB := rOutMin - rA * rInMin;
END_IF;

// Calculate the output
rOut :=  rA * rIn + rB;
```

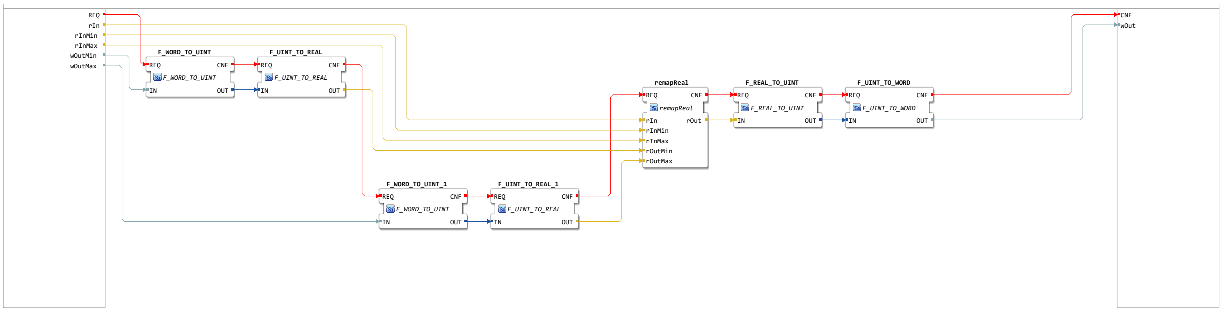

Normal Execution Request - Event REQ
CNF Event - Execution Confirmation
RemapReal2Word
1.0
- REAL rIn
wOut WORD -
- REAL rInMin
- REAL rInMax
- WORD wOutMin
- WORD wOutMax

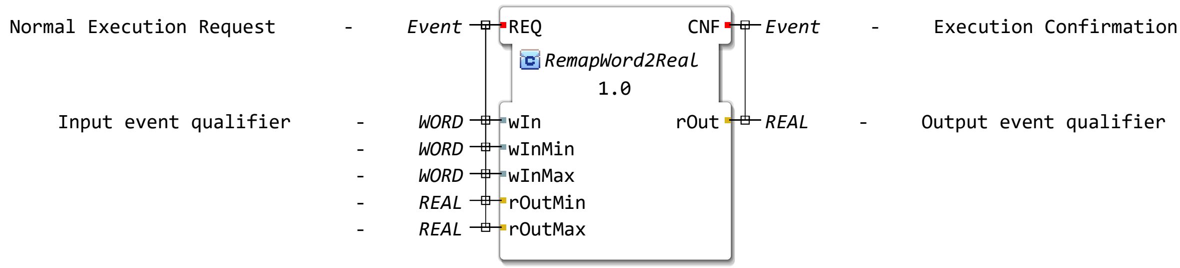
Inputs

Name	Type	Comment	Initial Value	Array Size
rIn	REAL			
rInMin	REAL			
rInMax	REAL			
wOutMin	WORD			
wOutMax	WORD			

Outputs

Name	Type	Comment	Initial Value	Array Size
wOut	WORD			


REQ
rIn
rInMin
rInMax
wOutMin
wOutMax
F_WORD_TO_UINT
REQ CNF
F_WORD_TO_UINT
IN OUT
F_UINT_TO_REAL
REQ CNF
F_UINT_TO_REAL
IN OUT
remapReal
REQ CNF
remapReal
rIn rOut
rInMin
rInMax
rOutMin
rOutMax
F_REAL_TO_UINT
REQ CNF
F_REAL_TO_UINT
IN OUT
F_UINT_TO_WORD
REQ CNF
F_UINT_TO_WORD
IN OUT
F_WORD_TO_UINT_1
REQ CNF
F_WORD_TO_UINT
IN OUT
F_UINT_TO_REAL_1
REQ CNF
F_UINT_TO_REAL
IN OUT
CNF
wOut

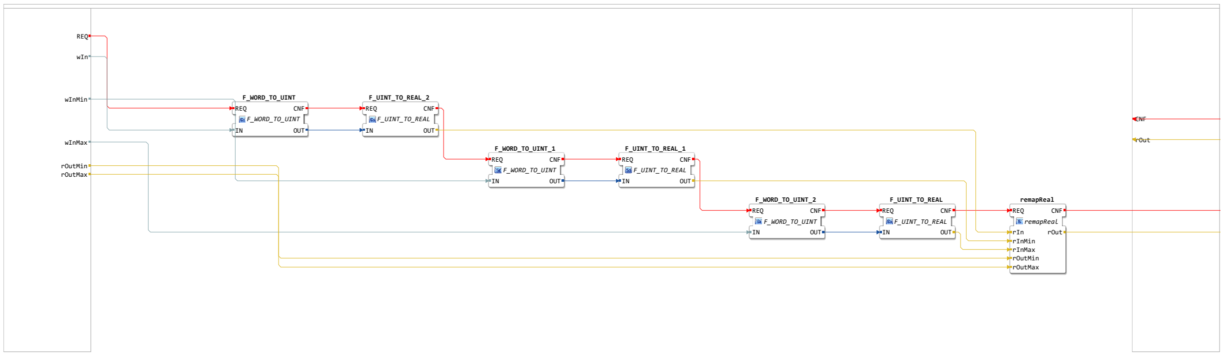

Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
RemapWord2Real
1.0
Input event qualifier - WORD
wIn
rOut
REAL - Output event qualifier
- WORD
wInMin
- WORD
wInMax
- REAL
rOutMin
- REAL
rOutMax

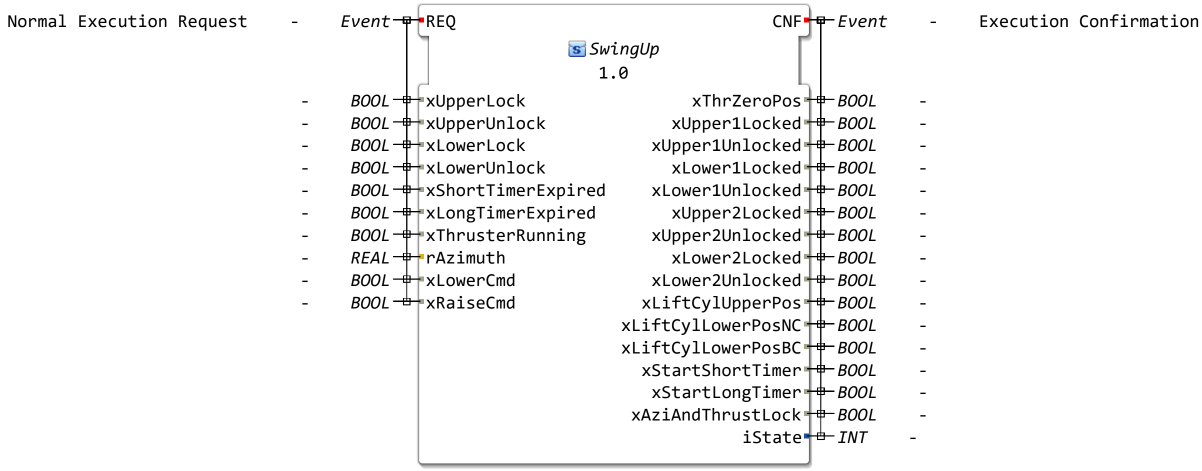
Inputs

Name	Type	Comment	Initial Value	Array Size
wIn	WORD	Input event...	0	
wInMin	WORD		16#0	
wInMax	WORD		16#0	
rOutMin	REAL		0.0	
rOutMax	REAL		0.0	

Outputs

Name	Type	Comment	Initial Value	Array Size
rOut	REAL	Output eve...		


REQ
wIn
wInMin
wInMax
rOutMin
rOutMax
F_WORD_TO_UINT
REQ CNF
F_WORD_TO_UINT
IN OUT
F_UINT_TO_REAL_2
REQ CNF
F_UINT_TO_REAL
IN OUT
F_WORD_TO_UINT_1
REQ CNF
F_WORD_TO_UINT
IN OUT
F_UINT_TO_REAL_1
REQ CNF
F_UINT_TO_REAL
IN OUT
F_WORD_TO_UINT_2
REQ CNF
F_WORD_TO_UINT
IN OUT
F_UINT_TO_REAL
REQ CNF
F_UINT_TO_REAL
IN OUT
remapReal
REQ CNF
remapReal
rIn rOut
rInMin
rInMax
rOutMin
rOutMax
CNF
rOut


Normal Execution Request - Event REQ
CNF Event - Execution Confirmation
SwingUp
1.0
- BOOL xUpperLock
- BOOL xUpperUnlock
- BOOL xLowerLock
- BOOL xLowerUnlock
- BOOL xShortTimerExpired
- BOOL xLongTimerExpired
- BOOL xThrusterRunning
- REAL rAzimuth
- BOOL xLowerCmd
- BOOL xRaiseCmd
xThrZeroPos BOOL -
xUpper1Locked BOOL -
xUpper1Unlocked BOOL -
xLower1Locked BOOL -
xLower1Unlocked BOOL -
xUpper2Locked BOOL -
xUpper2Unlocked BOOL -
xLower2Locked BOOL -
xLower2Unlocked BOOL -
xLiftCylUpperPos BOOL -
xLiftCylLowerPosNC BOOL -
xLiftCylLowerPosBC BOOL -
xStartShortTimer BOOL -
xStartLongTimer BOOL -
xAziAndThrustLock BOOL -
iState INT -

Inputs

Name	Type	Comment	Initial Value	Array Size
xUpperLock	BOOL			
xUpperUnlock	BOOL			
xLowerLock	BOOL			
xLowerUnlock	BOOL			
xShortTimerExpired	BOOL		FALSE	
xLongTimerExpired	BOOL		FALSE	
xThrusterRunning	BOOL			
rAzimuth	REAL			
xLowerCmd	BOOL			
xRaiseCmd	BOOL			

Outputs

Name	Type	Comment	Initial Value	Array Size
xThrZeroPos	BOOL		FALSE	
xUpper1Locked	BOOL		FALSE	
xUpper1Unlocked	BOOL		TRUE	
xLower1Locked	BOOL		TRUE	
xLower1Unlocked	BOOL		FALSE	
xUpper2Locked	BOOL		FALSE	
xUpper2Unlocked	BOOL		TRUE	
xLower2Locked	BOOL		TRUE	
xLower2Unlocked	BOOL		FALSE	
xLiftCylUpperPos	BOOL		FALSE	
xLiftCylLowerPosNC	BOOL		TRUE	
xLiftCylLowerPosBC	BOOL		TRUE	
xStartShortTimer	BOOL		FALSE	
xStartLongTimer	BOOL		FALSE	
xAziAndThrustLock	BOOL		FALSE	
iState	INT		4	

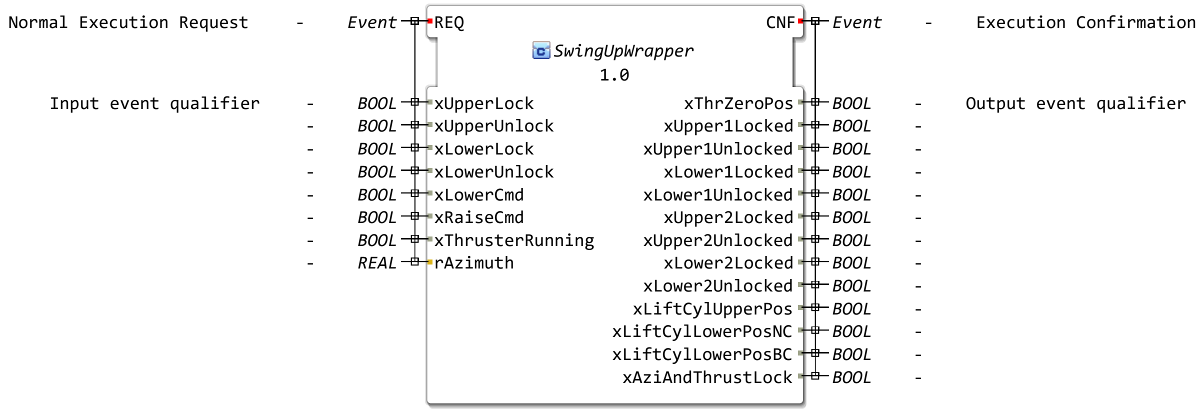
Internal Vars

Internal FBs

Name	Type	Comment	Initial Value	Array Size
iUpperLocked	INT		0	
iUpperUnlocking	INT		1	
iLowering	INT		2	
iLoweredLocking	INT		3	
iLoweredLocked	INT		4	
iLoweredUnlocking	INT		5	
iLifting	INT		6	
iUpperLocking	INT		7	

```
CASE iState OF

    0:      // Upper locked
        IF xUpperUnlock THEN
            xUpper1Locked := FALSE;
            xUpper2Locked := FALSE;
            xStartShortTimer := TRUE;
            iState := iUpperUnlocking;
        END_IF;

    1:      // Upper unlocking
        IF xShortTimerExpired THEN
            xUpper1Unlocked := TRUE;
            xUpper2Unlocked := TRUE;

            IF xUpperLock THEN
                xStartShortTimer := FALSE;
                xUpper1Unlocked := FALSE;
                xUpper2Unlocked := FALSE;
                iState := 7;
            END_IF;

            IF xLowerCmd THEN
                xStartShortTimer := FALSE;
                xStartLongTimer := TRUE;
                iState := iLowering;
            END_IF;

        END_IF;

    2:      // Lowering
        xLiftCylUpperPos := FALSE;
        xThrZeroPos := FALSE;

        IF xLongTimerExpired THEN
            xLiftCylLowerPosNC := TRUE;
            xLiftCylLowerPosBC := TRUE;

            IF xLowerLock THEN
                xLower1Unlocked := FALSE;
                xLower2Unlocked := FALSE;
                xStartLongTimer := FALSE;
                xStartShortTimer := TRUE;
                iState := iLoweredLocking;
            END_IF;
        END_IF;

    3:      // Lowered locking
        IF xShortTimerExpired THEN
            xLower1Locked := TRUE;
            xLower2Locked := TRUE;
            xAziAndThrustLock := FALSE;
            xStartShortTimer := FALSE;
            iState := iLoweredLocked;
        ELSE
            xStartShortTimer := TRUE;
        END_IF;

    4:      // Lowered locked
        IF xLowerUnlock AND NOT xThrusterRunning AND (rAzimuth < 0.17 OR rAzimuth > 6.113) THEN
            xLower1Locked := FALSE;
            xLower2Locked := FALSE;
            xAziAndThrustLock := TRUE;
            xStartShortTimer := TRUE;
            iState := iLoweredUnlocking;
        END_IF;

    5:      // Lowered unlocking
        IF xShortTimerExpired THEN
            xLower1Unlocked := TRUE;
            xLower2Unlocked := TRUE;

```

```
            IF xLowerLock THEN
                xStartShortTimer := FALSE;
                xLower1Unlocked := FALSE;
                xLower2Unlocked := FALSE;
                iState := 3;
            END_IF;

            IF xRaiseCmd THEN
                xStartShortTimer := FALSE;
                xStartLongTimer := TRUE;
                iState := iLifting;
            END_IF;
        END_IF;

    6:      // Lifting
        xLiftCylLowerPosNC := FALSE;
        xLiftCylLowerPosBC := FALSE;

        IF xLongTimerExpired THEN
            xLiftCylUpperPos := TRUE;

            IF xUpperLock THEN
                xUpper1Unlocked := FALSE;
                xUpper2Unlocked := FALSE;
                xStartShortTimer := TRUE;
                xStartLongTimer := FALSE;
                iState := iUpperLocking;
            END_IF;
        END_IF;

    7:      // Upper Locking
        IF xShortTimerExpired THEN
            xUpper1Locked := TRUE;
            xUpper2Locked := TRUE;
            xThrZeroPos := TRUE;
            xStartShortTimer := FALSE;
            iState := iUpperLocked;
        ELSE
            xStartShortTimer := TRUE;
        END_IF;

END_CASE;
```

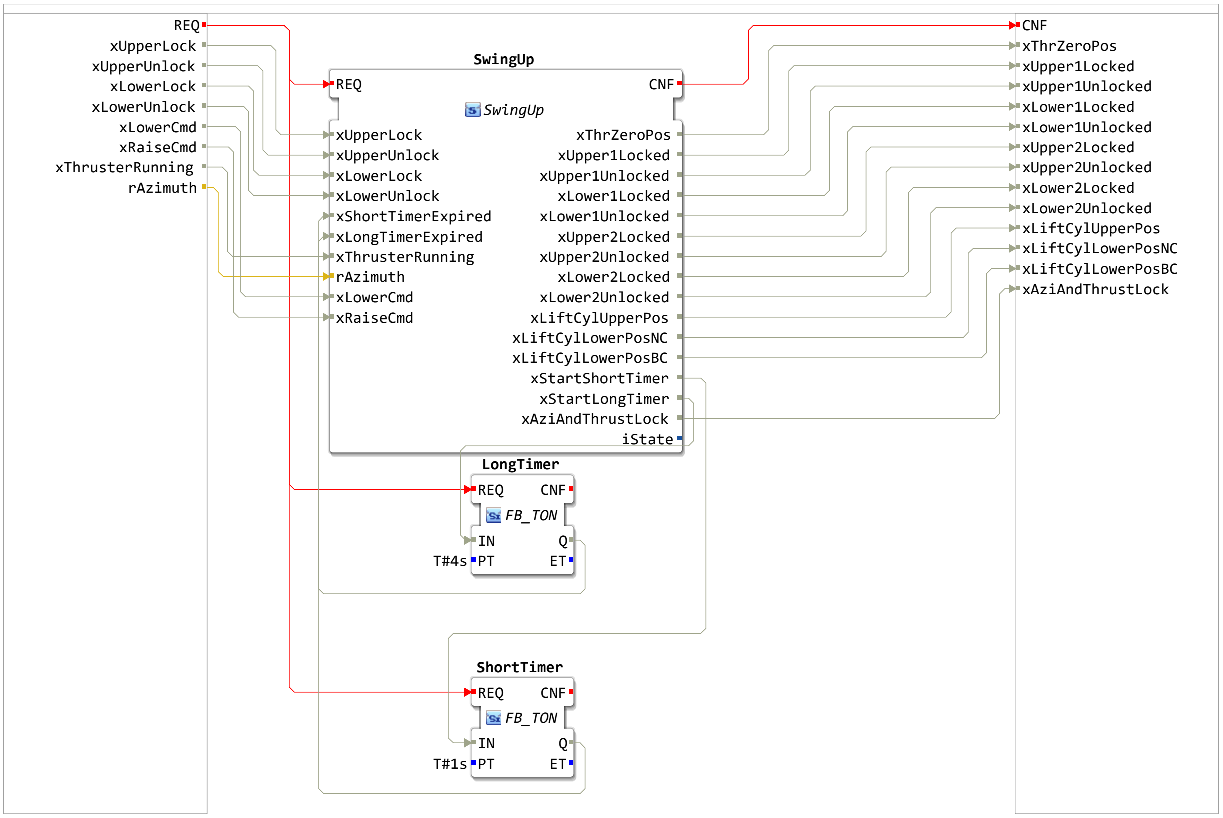

Normal Execution Request - Event REQ
CNF Event - Execution Confirmation
SwingUpWrapper
1.0
Input event qualifier - BOOL xUpperLock
- BOOL xUpperUnlock
- BOOL xLowerLock
- BOOL xLowerUnlock
- BOOL xLowerCmd
- BOOL xRaiseCmd
- BOOL xThrusterRunning
- REAL rAzimuth
xThrZeroPos BOOL - Output event qualifier
xUpper1Locked BOOL -
xUpper1Unlocked BOOL -
xLower1Locked BOOL -
xLower1Unlocked BOOL -
xUpper2Locked BOOL -
xUpper2Unlocked BOOL -
xLower2Locked BOOL -
xLower2Unlocked BOOL -
xLiftCylUpperPos BOOL -
xLiftCylLowerPosNC BOOL -
xLiftCylLowerPosBC BOOL -
xAziAndThrustLock BOOL -

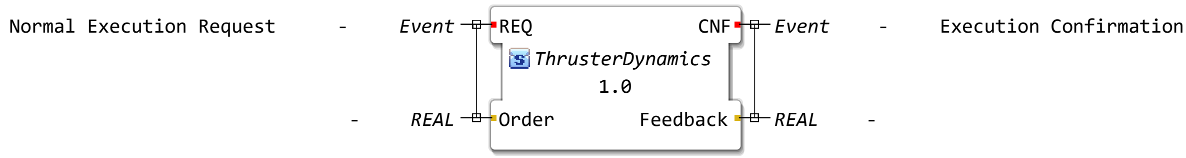
Inputs

Name	Type	Comment	Initial Value	Array Size
xUpperLock	BOOL	Input event...		
xUpperUnlock	BOOL			
xLowerLock	BOOL			
xLowerUnlock	BOOL			
xLowerCmd	BOOL			
xRaiseCmd	BOOL			
xThrusterRunning	BOOL			
rAzimuth	REAL			

Outputs

Name	Type	Comment	Initial Value	Array Size
xThrZeroPos	BOOL	Output eve...		
xUpper1Locked	BOOL			
xUpper1Unlocked	BOOL			
xLower1Locked	BOOL			
xLower1Unlocked	BOOL			
xUpper2Locked	BOOL			
xUpper2Unlocked	BOOL			
xLower2Locked	BOOL			
xLower2Unlocked	BOOL			
xLiftCylUpperPos	BOOL			
xLiftCylLowerPosNC	BOOL			
xLiftCylLowerPosBC	BOOL			
xAziAndThrustLock	BOOL			


REQ
xUpperLock
xUpperUnlock
xLowerLock
xLowerUnlock
xLowerCmd
xRaiseCmd
xThrusterRunning
rAzimuth
SwingUp
REQ
CNF
SwingUp
xUpperLock
xUpperUnlock
xLowerLock
xLowerUnlock
xShortTimerExpired
xLongTimerExpired
xThrusterRunning
rAzimuth
xLowerCmd
xRaiseCmd
xThrZeroPos
xUpper1Locked
xUpper1Unlocked
xLower1Locked
xLower1Unlocked
xUpper2Locked
xUpper2Unlocked
xLower2Locked
xLower2Unlocked
xLiftCylUpperPos
xLiftCylLowerPosNC
xLiftCylLowerPosBC
xStartShortTimer
xStartLongTimer
xAziAndThrustLock
iState
LongTimer
REQ
CNF
FB_TON
IN
Q
T#4s
PT
ET
ShortTimer
REQ
CNF
FB_TON
IN
Q
T#1s
PT
ET
CNF
xThrZeroPos
xUpper1Locked
xUpper1Unlocked
xLower1Locked
xLower1Unlocked
xUpper2Locked
xUpper2Unlocked
xLower2Locked
xLower2Unlocked
xLiftCylUpperPos
xLiftCylLowerPosNC
xLiftCylLowerPosBC
xAziAndThrustLock


Normal Execution Request - Event REQ
CNF Event - Execution Confirmation
ThrusterDynamics
1.0
- REAL Order
Feedback REAL -

Inputs

Name	Type	Comment	Initial Value	Array Size
Order	REAL			

Outputs

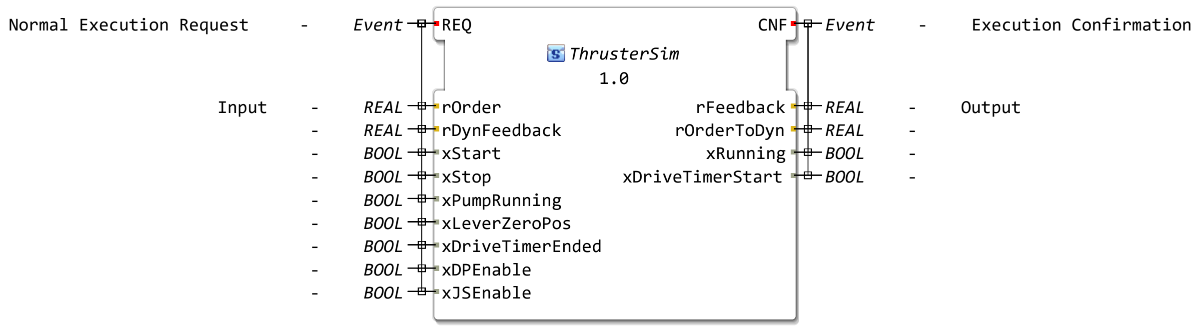
Name	Type	Comment	Initial Value	Array Size
Feedback	REAL			

Internal Vars

Internal FBs

Name	Type	Comment	Initial Value	Array Size
Integrator_DSTATE	LREAL			
Filter_DSTATE	LREAL			
Delay_DSTATE	LREAL			
rtb_Sum_p	LREAL			
rtb_FilterCoefficient	LREAL			
rtb_Sum	LREAL			
rtb_SquareRoot	LREAL			
DCMOTOR1_tmp	LREAL			
temp1	LREAL			
DCMOTOR1_states	LREAL			

```
(* Outputs for Atomic SubSystem: '<Root>/Subsystem Discrete version 1' *)
(* Sum: '<S2>/Sum' incorporates:
 *  Delay: '<S2>/Delay' *)
rtb_Sum_p := Order - Delay_DSTATE;
(* Gain: '<S39>/Filter Coefficient' incorporates:
 *  DiscreteIntegrator: '<S31>/Filter'
 *  Gain: '<S30>/Derivative Gain'
 *  Sum: '<S31>/SumD' *)
rtb_FilterCoefficient := ((1.1 * rtb_Sum_p) - Filter_DSTATE) * 0.952380952380952;
(* DiscreteTransferFcn: '<S2>/DC MOTOR1' incorporates:
 *  DiscreteIntegrator: '<S36>/Integrator'
 *  Gain: '<S41>/Proportional Gain'
 *  Sum: '<S45>/Sum' *)
DCMOTOR1_tmp := (((2.0 * rtb_Sum_p) + Integrator_DSTATE) + rtb_FilterCoefficient) -
(-0.951219512195122 * DCMOTOR1_states);
(* DiscreteTransferFcn: '<S2>/PROPELLER DYNAMICS' incorporates:
 *  DiscreteTransferFcn: '<S2>/DC MOTOR1'
 *  Gain: '<S2>/GEAR RATIO' *)
rtb_SquareRoot := (((59.268292682926827 * DCMOTOR1_tmp) + (59.268292682926827 *
DCMOTOR1_states)) * 0.16583333333333333) * 0.000181413307416;
(* Switch: '<S2>/Switch' incorporates:
 *  Constant: '<S2>/Constant'
 *  Constant: '<S2>/Constant1'
 *  Constant: '<S3>/Constant'
 *  RelationalOperator: '<S3>/Compare' *)
IF rtb_SquareRoot <= 0.0 THEN
    rtb_Sum := -1.0;
ELSE
    rtb_Sum := 1.0;
END_IF;
(* End of Switch: '<S2>/Switch' *)

(* DiscreteTransferFcn: '<S2>/FEEDBACK' incorporates:
 *  Abs: '<S2>/Abs'
 *  Gain: '<S2>/RPS to RPM'
 *  Product: '<S2>/Product1'
 *  Sqrt: '<S2>/Square Root' *)
temp1 := ABS(rtb_SquareRoot);
Feedback := ((rtb_Sum * SQRT(temp1)) * 60.0) * 0.50251256281407;
(* Update for Delay: '<S2>/Delay' *)
Delay_DSTATE := Feedback;
(* Update for DiscreteIntegrator: '<S36>/Integrator' incorporates:
 *  Gain: '<S33>/Integral Gain' *)
Integrator_DSTATE := ((0.89 * rtb_Sum_p) * 0.1) + Integrator_DSTATE;
(* Update for DiscreteIntegrator: '<S31>/Filter' *)
Filter_DSTATE := (0.1 * rtb_FilterCoefficient) + Filter_DSTATE;
(* Update for DiscreteTransferFcn: '<S2>/DC MOTOR1' *)
DCMOTOR1_states := DCMOTOR1_tmp;
(* End of Outputs for SubSystem: '<Root>/Subsystem Discrete version 1' *)
```


Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
ThrusterSim
1.0
Input - REAL rOrder
- REAL rDynFeedback
- BOOL xStart
- BOOL xStop
- BOOL xPumpRunning
- BOOL xLeverZeroPos
- BOOL xDriveTimerEnded
- BOOL xDPEnable
- BOOL xJSEnable
rFeedback REAL - Output
rOrderToDyn REAL -
xRunning BOOL -
xDriveTimerStart BOOL -

Inputs

Name	Type	Comment	Initial Value	Array Size
rOrder	REAL	Input	0	
rDynFeedback	REAL		0	
xStart	BOOL			
xStop	BOOL			
xPumpRunning	BOOL			
xLeverZeroPos	BOOL			
xDriveTimerEnded	BOOL			
xDPEnable	BOOL			
xJSEnable	BOOL			

Outputs

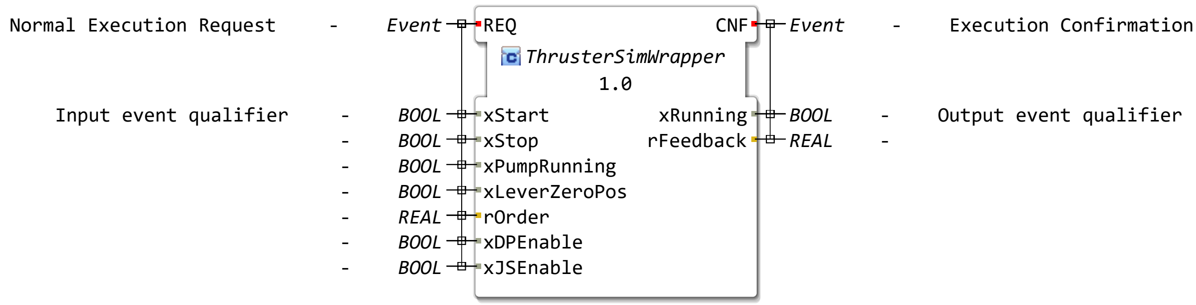
Name	Type	Comment	Initial Value	Array Size
rFeedback	REAL	Output	0	
rOrderToDyn	REAL			
xRunning	BOOL		0	
xDriveTimerStart	BOOL		0	

Internal Vars

Internal FBs

Name	Type	Comment	Initial Value	Array Size
xThrusterRunning	BOOL			
rFinalOrder	REAL	Order value used in the dynamics of the thruster	0	

```
// Start the thruster
IF xStop THEN
    // TODO: Should the thruster stop if the pump signal is lost???
    //TODO: Add the stop dynamics for the thruster
    xDriveTimerStart := FALSE;
    xRunning := FALSE;

ELSIF xPumpRunning AND xStart THEN
    // TODO: Add the start dynamics for the thruster
    xDriveTimerStart := TRUE;

END_IF;

// Start drive
IF xDriveTimerEnded THEN
    xRunning := TRUE;
END_IF;


// Set the RPM order for the thruster
IF xRunning AND (xJSEnable OR xDPEnable) THEN
    rOrderToDyn := rOrder;

ELSIF NOT xRunning OR xLeverZeroPos THEN
    rOrderToDyn := 0;

ELSIF xRunning THEN
    rOrderToDyn := rOrder;

ELSE
rOrderToDyn := rOrderToDyn;
END_IF;
// Simulate the thruster dynamics


rFeedback := rDynFeedback;
```

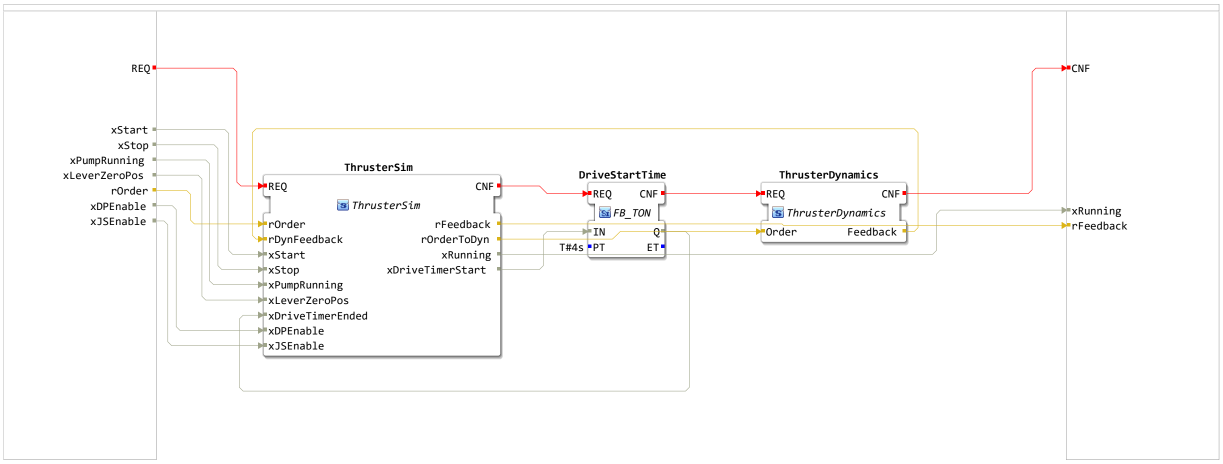

Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
ThrusterSimWrapper
1.0
Input event qualifier - BOOL xStart
xRunning BOOL - Output event qualifier
- BOOL xStop
rFeedback REAL -
- BOOL xPumpRunning
- BOOL xLeverZeroPos
- REAL rOrder
- BOOL xDPEnable
- BOOL xJSEnable

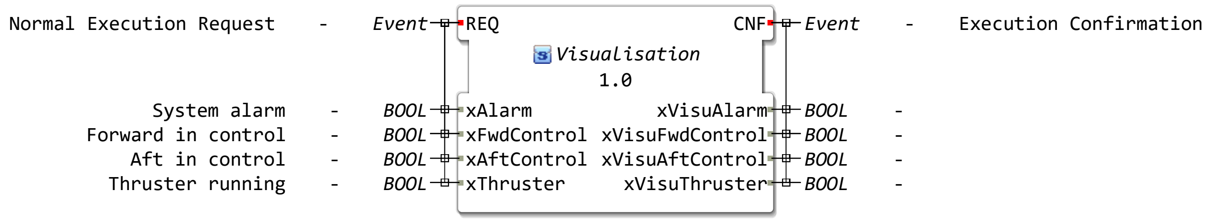
Inputs

Name	Type	Comment	Initial Value	Array Size
xStart	BOOL	Input event...	0	
xStop	BOOL		0	
xPumpRunning	BOOL		0	
xLeverZeroPos	BOOL		0	
rOrder	REAL		0	
xDPEnable	BOOL		0	
xJSEnable	BOOL		0	

Outputs

Name	Type	Comment	Initial Value	Array Size
xRunning	BOOL	Output eve...	0	
rFeedback	REAL		0	


REQ
xStart
xStop
xPumpRunning
xLeverZeroPos
rOrder
xDPEnable
xJSEnable
ThrusterSim
REQ
CNF
ThrusterSim
rOrder
rDynFeedback
xStart
xStop
xPumpRunning
xLeverZeroPos
xDriveTimerEnded
xDPEnable
xJSEnable
rFeedback
rOrderToDyn
xRunning
xDriveTimerStart
DriveStartTime
REQ
CNF
FB_TON
IN
Q
T#4s
PT
ET
ThrusterDynamics
REQ
CNF
ThrusterDynamics
Order
Feedback
CNF
xRunning
rFeedback


Normal Execution Request - Event
REQ
CNF
Event - Execution Confirmation
Visualisation
1.0
System alarm - BOOL
xAlarm
xVisuAlarm
BOOL -
Forward in control - BOOL
xFwdControl
xVisuFwdControl
BOOL -
Aft in control - BOOL
xAftControl
xVisuAftControl
BOOL -
Thruster running - BOOL
xThruster
xVisuThruster
BOOL -

Inputs

Name	Type	Comment	Initial Value	Array Size
xAlarm	BOOL	System alar...		
xFwdControl	BOOL	Forward in ...		
xAftControl	BOOL	Aft in contr...		
xThruster	BOOL	Thruster ru...		

Outputs

Name	Type	Comment	Initial Value	Array Size
xVisuAlarm	BOOL			
xVisuFwdControl	BOOL			
xVisuAftControl	BOOL			
xVisuThruster	BOOL			

```
    xVisuAlarm := xAlarm;
    xVisuFwdControl := xFwdControl;
    xVisuAftControl :=  xAftControl;
    xVisuThruster := xThruster;
```

J Docker source code

This appendix shows the docker source code for the Dockerfile and the accompanying bash-script.
Includes:

- Dockerfile (2 pages)
- Bash-script (6 pages)

File - Dockerfile_forte

```
#*******************************************************************************************
# Copyright (c) 2020 fortiss GmbH
# This program and the accompanying materials are made available under the
# terms of the Eclipse Public License 2.0 which is available at
# http://www.eclipse.org/legal/epl-2.0.
#
# SPDX-License-Identifier: EPL-2.0
#
# Contributors:
#     Jose Cabral
#       -  initial implementation
#     Espen Godø
#       -  Changed the setup to match the  needs of our system
# ******************************************************************************************

FROM alpine:edge as forteBuilder
MAINTAINER Jose Cabral <cabralcochi@gmail.com>

RUN apk add --no-cache bash \
    ca-certificates \
    cmake \
    g++ \
    git \
    gcc \
    make \
    python3 \
    py-pip \
    py-six \
    openssh-client \
    subversion

#COPY ./docker/scripts/prepareOtherRepos.sh /usr/scripts_temp/
#COPY ./docker/scripts/commonFunctions.sh /usr/scripts_temp/
#RUN /usr/scripts_temp/prepareOtherRepos.sh -r -o

ARG FORTE_USER_FLAGS=""
ENV FORTE_USER_FLAGS $FORTE_USER_FLAGS

ARG FORTE_EXTERNAL_MODULES=""
ENV FORTE_EXTERNAL_MODULES $FORTE_EXTERNAL_MODULES

```

```
COPY . /usr/repos/org.eclipse.4diac.forte/
RUN /usr/repos/org.eclipse.4diac.forte/docker/scripts/buildForte_test.sh


FROM alpine:edge
WORKDIR /usr/forte_output/rpi/
COPY --from=forteBuilder /usr/forte_output/rpi/* ./
RUN apk add --no-cache libstdc++

ENV LD_LIBRARY_PATH /usr/forte_output/rpi/

ARG FORTE_BOOT_FILE=/usr/forte_boot/forte.fboot
ENV FORTE_BOOT_FILE $FORTE_BOOT_FILE

ARG FORTE_EXTRA_OPTIONS=""
ENV FORTE_EXTRA_OPTIONS $FORTE_EXTRA_OPTIONS

EXPOSE 4840 4841 4842 61499 61500 61501 61502 61503 61504 61505

# Expose the ports used for communication between IEC61499 devices
EXPOSE 61000/udp 61001/udp 61002/udp 61003/udp 61004/udp 61005/udp 61006/udp 61007/udp
# EXPOSE 61000-62000/udp

CMD /usr/forte_output/rpi/forte -f $FORTE_BOOT_FILE $FORTE_EXTRA_OPTIONS

```

File - buildForte_opc.sh

```
#!/bin/bash
#***********************************************************************************************
# Copyright (c) 2020 fortiss GmbH
# This program and the accompanying materials are made available under the
# terms of the Eclipse Public License 2.0 which is available at
# http://www.eclipse.org/legal/epl-2.0.
#
# SPDX-License-Identifier: EPL-2.0
#
# Contributors:
#      Jose Cabral
#        -  initial implementation
#      Espen Godø
#        -  Refactored the setup to match the needs of our system
# ******************************************************************************************

# exit when any command fails
set -e

# store the the last executed command
trap 'lastCommand=$currentCommand; currentCommand=$BASH_COMMAND' DEBUG
# echo an error message before exiting
trap 'if [ $? -eq 0 ]; then echo \"The script exited without errors\"; else echo "\"${lastCommand}\" command failed with exit code $?."; fi' EXIT


BASH_SCRIPT_PATH="$(cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"

#source "${BASH_SCRIPT_PATH}"/prepareOtherRepos.sh "$@"

#####################################################################
#####################  Variables ###################################
#####################################################################
# - FORTE_REPO_PATH: The forte repository is set to the parent folder of this script. If the script is executed to another repository,
# set the FORTE_REPO_PATH variable to the absolut path of
```

File - buildForte_opc.sh

```
the desired forte repository before calling this script
# - FORTE_OUTPUT_DIR: The output directory where all devices folders with their binaries will be placed. Default /usr/forte_output/

if [ -z "$FORTE_REPO_PATH" ]; then
  FORTE_REPO_PATH="$(cd ${BASH_SCRIPT_PATH} && cd ../../ && pwd )/"
fi

if [ -z "$FORTE_OUTPUT_DIR" ]; then
  FORTE_OUTPUT_DIR=/usr/forte_output/
fi

COMMON_FLAGS="-DCMAKE_BUILD_TYPE=Debug \
              -DFORTE_LOGLEVEL=Debug \
              -DFORTE_BUILD_EXECUTABLE=ON \
              -DFORTE_COM_ETH=ON \
              -DFORTE_MODULE_UTILS=ON \
              -DFORTE_MODULE_CONVERT=ON \
              -DFORTE_MODULE_IEC61131=ON \
              -DFORTE_COM_HTTP=ON \
              -DFORTE_COM_HTTP_LISTENING_PORT=8080 \
              ${USER_FLAGS} \
              -DFORTE_EXTERNAL_MODULES_DIRECTORY=${FORTE_EXTERNAL_MODULES}"

#ARM_COMPILER="-DCMAKE_C_COMPILER=${ARM_BIN}/arm-linux-gnueabihf-gcc \
#              -DCMAKE_CXX_COMPILER=${ARM_BIN}/arm-linux-gnueabihf-g++ \
#              -DCMAKE_FIND_ROOT_PATH=${ARM_ROOT} "

####################################################################
#####################  FUNCTIONS ###################################
####################################################################

##################################
# Print message in a nice format #
#   $1: message to be printed    #
##################################
function printMessage(){
  SIZE_OF_MESSAGE=$((${#1} + 4))
  WRAPPER=$(printf '#%.0s' $(eval "echo {1.."$(($SIZE_OF_MESSAGE))"}"))
```

File - buildForte_opc.sh

```
  echo -e "\n$WRAPPER"
  printf '# %s #\n' "$1"
  echo -e "$WRAPPER\n"
}

##################################
# Clean folder and go inside it #
#   $1: Folder to clean          #
##################################
function cleanFolderAndGoThere(){
  if [ -d "${1}" ];
  then
     rm -r "${1}"
  fi

  mkdir -p "${1}"
  cd "${1}"
}

##################################
# Compile mbedtls                #
#   - $1: architecture to build #
##################################
function compileMbetls(){
  if [ ${CLEAN_ALL} -eq "1" ]  || ! [ -f "${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${1}/library/libmbedcrypto.a" ]; then

    printMessage "Compiling mbedtls for ${1}"

    cleanFolderAndGoThere "${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${1}"

    cmake -DENABLE_TESTING=Off \
          ${CMAKE_EXTRA_FLAGS} \
          "${REPOS_DIR}${MBEDTLS_FOLDER_NAME}" && \
      make -j8
  else
     printMessage "Library for mbedtls for $1 already found and won't be recompiled"
  fi
}

##################################
# Compile open62541              #
#   - $1: architecture to build #
##################################
function compileOpen62541(){
```

File - buildForte_opc.sh

```
  if [ ${CLEAN_ALL} -eq "1" ] || ! [ -f "${REPOS_DIR}${OPCUA_FOLDER_NAME}/bin/${1}/bin/libopen62541.a" ]; then
    printMessage "Compiling open62541 for ${1}"

    cleanFolderAndGoThere "${REPOS_DIR}${OPCUA_FOLDER_NAME}/bin/${1}"

    cmake -DBUILD_SHARED_LIBS=OFF \
          -DCMAKE_BUILD_TYPE=RelWithDebInfo \
          -DUA_ENABLE_AMALGAMATION=ON \
          -DUA_ENABLE_ENCRYPTION=ON \
          -DMBEDCRYPTO_LIBRARY="${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${1}/library/libmbedcrypto.a" \
          -DMBEDTLS_LIBRARY="${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${1}/library/libmbedtls.a" \
          -DMBEDX509_LIBRARY="${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${1}/library/libmbedx509.a" \
          -DMBEDTLS_INCLUDE_DIRS="${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/include" \
          ${CMAKE_EXTRA_FLAGS} \
          ${REPOS_DIR}${OPCUA_FOLDER_NAME} && \
      make -j8
  else
     printMessage "Library for open62541 for $1 already found and won't be recompiled"
  fi
}

# The function takes 3 parameters:
#    - the architecture (linux32, arm, ...) used for the needed libraries
#    - the device (linux32, rpi, BBB, ...) used for the exported folder under ${FORTE_OUTPUT_DIR}
#    - extra parameteters to be added to CMake
function compile(){

  CURRENT_ARCH=$1
  CURRENT_DEVICE=$2
  EXTRA_FLAGS=$3

  CURRENT_BIN="${FORTE_REPO_PATH}bin/${CURRENT_DEVICE}"
  CURRENT_OUTPUT="${FORTE_OUTPUT_DIR}${CURRENT_DEVICE}"

  OPCUA_FLAGS="-DFORTE_COM_OPC_UA=ON \
               -DFORTE_COM_OPC_UA_INCLUDE_DIR=${REPOS_DIR}${OPCUA_FOLDER_NAME}/bin/${CURRENT_ARCH}/ \
               -DFORTE_COM_OPC_UA_LIB_DIR=${REPOS_DIR}${OPCUA_FOLDER_NAME}/bin/${CURRENT_ARCH}/bin \
```

File - buildForte_opc.sh

```
                  -DFORTE_COM_OPC_UA_LIB=libopen62541.a \
                  -DFORTE_COM_OPC_UA_MASTER_BRANCH=ON \
                  -DFORTE_COM_OPC_UA_ENCRYPTION=ON \
                  -DFORTE_COM_OPC_UA_ENCRYPTION_INCLUDE_DIR=${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/include \
                  -DFORTE_COM_OPC_UA_ENCRYPTION_LIB_DIR=${REPOS_DIR}${MBEDTLS_FOLDER_NAME}/bin/${CURRENT_ARCH}/library/"

  ALL_FLAGS="${COMMON_FLAGS} ${OPCUA_FLAGS} ${EXTRA_FLAGS}"

  printMessage "Compiling forte for arch ${CURRENT_ARCH} and device ${CURRENT_DEVICE}"
  echo -e "\nThe following flags will be used: \n${ALL_FLAGS}\n"

  cleanFolderAndGoThere "${CURRENT_BIN}"

  cmake ${ALL_FLAGS} ${FORTE_REPO_PATH}
  make -j8

  mkdir -p "${CURRENT_OUTPUT}"
  cp src/forte "${CURRENT_OUTPUT}"

#  if [ "${LUA_ENABLED}" -eq "1" ]; then
#    cp "${CURRENT_LUA_LIBS}"*.so* "${CURRENT_OUTPUT}"
#  fi
}

### Prepare OPC

###################
# Prepare mbedtls #
###################

REPOS_DIR=/usr/repos/
MBEDTLS_FOLDER_NAME=mbedtls
OPCUA_FOLDER_NAME=open62541
ARM_ARCH=arm


CURRENT_FOLDER="${REPOS_DIR}${MBEDTLS_FOLDER_NAME}"

if ! [ -d "${CURRENT_FOLDER}" ]
then
```

File - buildForte_opc.sh

```
  printMessage "Creating folder for mbetdls repository"

  mkdir -p "${CURRENT_FOLDER}" && \
  git clone https://github.com/ARMmbed/mbedtls "${CURRENT_FOLDER}"
fi

cd "${CURRENT_FOLDER}" && \
git checkout mbedtls-2.7.1

#####################
# Prepare open62541 #
#####################

CURRENT_FOLDER="${REPOS_DIR}${OPCUA_FOLDER_NAME}"

if ! [ -d "${CURRENT_FOLDER}" ]
then
  printMessage "Creating folder for opcua repository"

  mkdir -p "${CURRENT_FOLDER}" && \
  git clone https://github.com/open62541/open62541.git "${CURRENT_FOLDER}"
fi

cd "${CURRENT_FOLDER}" && \
git checkout v1.0 && \
git submodule update --recursive --init


cd "${FORTE_REPO_PATH}"


####################################################################
################# COMPILE FOR RPI #################################
####################################################################
compileMbetls "${ARM_ARCH}"
compileOpen62541 "${ARM_ARCH}"

RPI_FLAGS="-DFORTE_ARCHITECTURE=Posix \
       -DCMAKE_CXX_FLAGS=-static-libstdc++"

compile "${ARM_ARCH}" "rpi" "${RPI_FLAGS}"

```

K Startup script

This appendix shows the source code for the start up script used on the PLC.

Includes:

- Dockerfile (1 page)

File - auto_start.sh

```
#!/bin/sh
##
# Auto start script
# This program was used to autostart the docker container and the local Forte
##

# Remove the last used docker container
docker stop forte_opc
docker container rm forte_opc

# start the docker container
docker run -d \
--net=host \
--name forte_opc \
-v /home/admin/forte-boot:/app/ \
--env FORTE_EXTRA_OPTIONS="-c localhost:61530" \
--env FORTE_BOOT_FILE=/app/thrusterSimulator_A200_OPC.fboot \
forte:latest

# Start the forte version that uses the k-bus and IO
/usr/bin/forte \
  -c localhost:61520 \
  -f /home/admin/forte-boot/thrusterSimulator_A200.fboot

```

L Initial model Matlab/Simulink

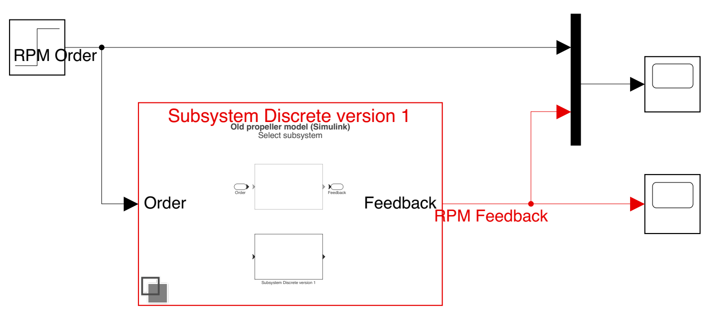
This appendix shows the Matlab program and the Simulink model used for the initial propeller model.
Includes:

- Matlab script (1 page)
- Simulink model (6 pages)

```
%%
% Old propeller model (MATLB)
% This is a copy of the program used by Skarbø et al. without any changes
% This program is only used to set the parameters for the Simulink model
%% DC MOTOR-----------------------------------------------------------------------------------------
%1.st order system used to simulate/approximate a DC motor.
Km = 2430; % Torque constant, based of torque produced from datasheet.
Tm = 2; % Time constant, based off of settling time of 8 sec.
TF_motor = tf(Km, [Tm 1])
%To check magnitude and time constant.
%stepplot(TF_motor) %test to see if system behaves as intended
%bode(TF_motor);
%% PROPELLER DYNAMICS---------------------------------------------------------------------------------
%calculates the torque and thrust coefficients based on the parameters
%given.
%Max rpm of motor and propeller. Used to calculate gear ratio.
max_RPM_motor = 1200; % max RPM of motor
max_RPM_propeller = 199; % max RPM of propeller
gear_ratio = max_RPM_propeller/max_RPM_motor;
%creates a torque and thrust coefficient based on the values
%from the Wagening B screw series propeller table
%The propeller characteristics
D = 2.4;
z = 4;
rho = 1025; %(1020-2019)
AEAO = 0.7; %propeller blade divided by the
PD = 1; Ja = 0;
% diameter of propeller in meters
% number of blades on propeller
% average density of surface seawater
% blade area ratio. The area of the circular area of the propeller rotation
% Pitch diameter ratio
% Avdvanced velocity
[KT, KQ] = wageningen(Ja,PD,AEAO,z); % Function call on wageningen.m made by Thor I. Fossen
```

FPP_Simulink_PID_remake

Old propeller model (Simulink)

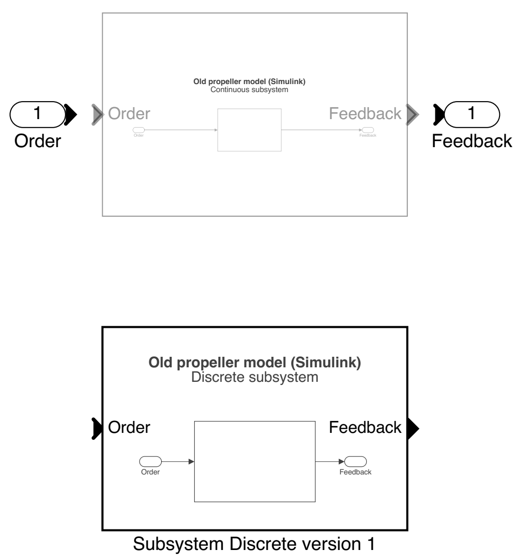


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43 | page 1/6

FPP_Simulink_PID_remake/Subsystem

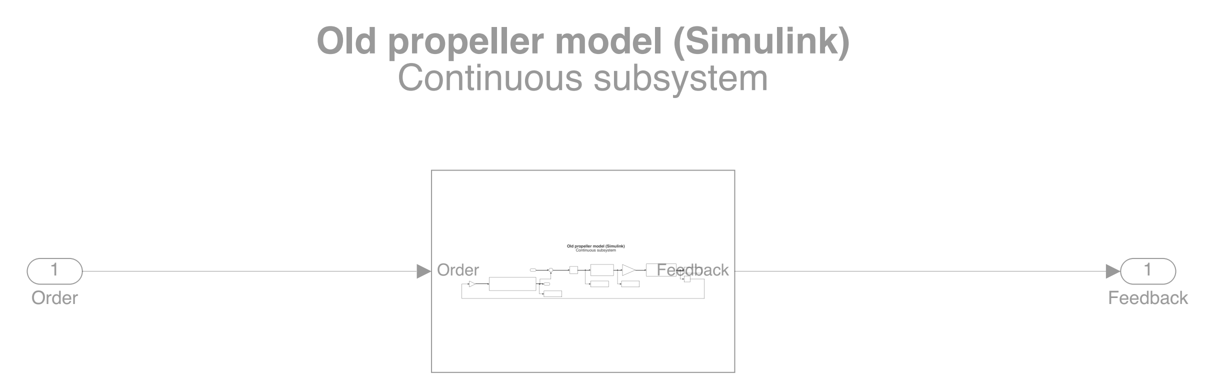
Old propeller model (Simulink)
Select subsystem



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43 | page 2/6

FPP_Simulink_PID_remake/Subsystem/Subsystem

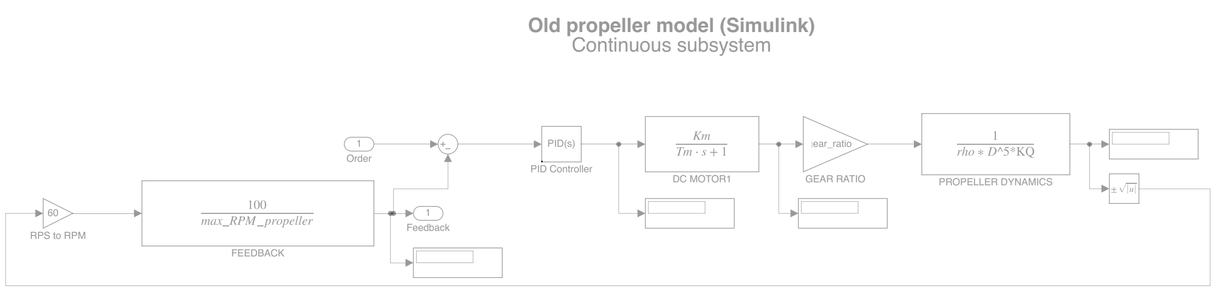


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43

page 3/6

FPP_Simulink_PID_remake/Subsystem/Subsystem/Subsystem

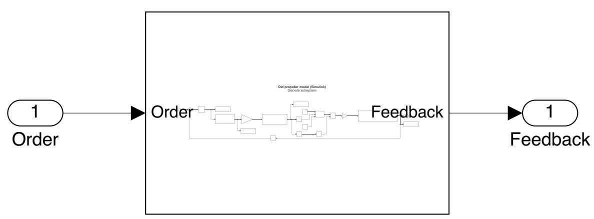


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43 | page 4/6

FPP_Simulink_PID_remake/Subsystem/Subsystem Discrete version 1

Old propeller model (Simulink)
Discrete subsystem



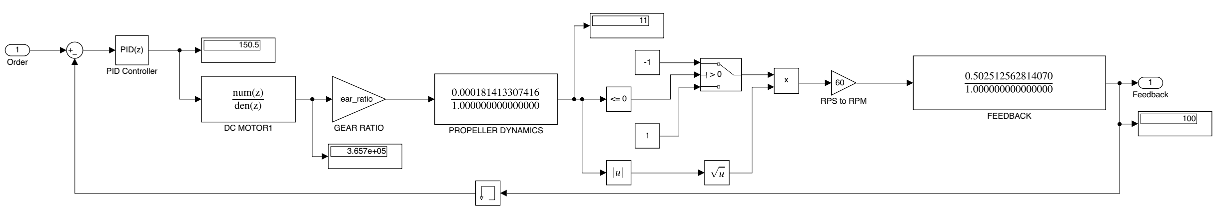
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43

page 5/6

FPP_Simulink_PID_remake/Subsystem/Subsystem Discrete version 1/Subsystem

Old propeller model (Simulink)
Discrete subsystem



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/FPP_old/FPP_Simulink_PID_remake.slx

printed 20-May-2022 00:43 | page 6/6

M Propeller model Matlab/Simulink

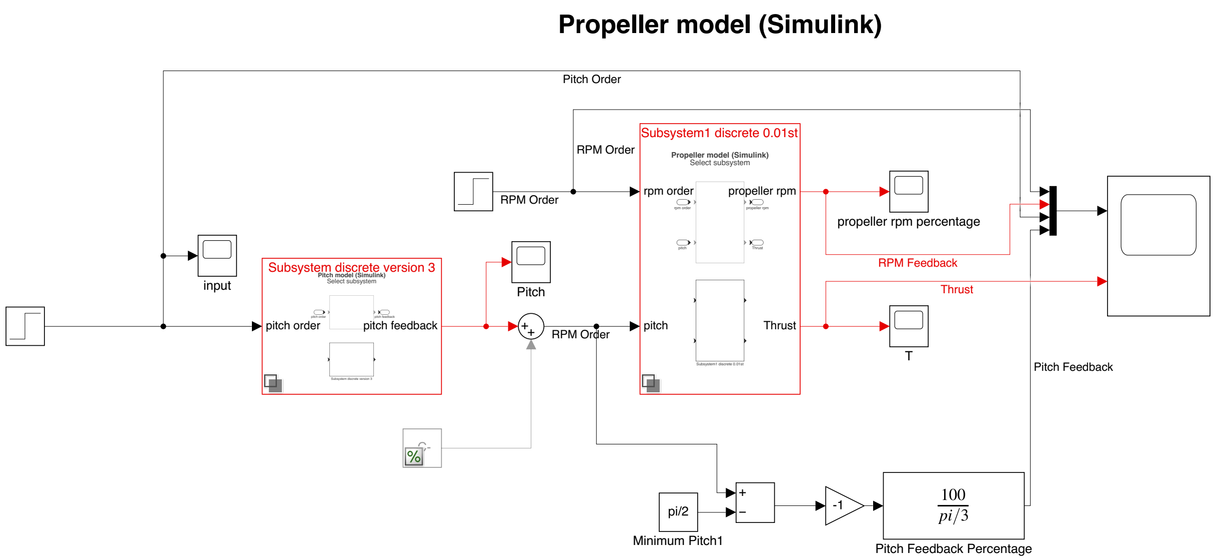
This appendix shows the Matlab program and the Simulink model used for the new propeller model.

- Matlab script (1 page)
- Simulink model (19 pages)

Note pages 9-13 and 15-19 in the Simulink model describe the function blocks used in the subsystems.

```
%%
% CPP Model (MATLAB)
% Implementation of the model described by Healy et al.
%%
clear

rho = 1025;          % water density
D = 2;               % propeller diameter in meter
R = D/2;             % propeller radius in meter
A = 3.14 * R^2;      % tunnel cross-sectional area
L = 2.5;             % tunnel length
gamma = 0.5;         % effective added mass ratio
delta_beta = 0.2;    % momentum coefficient

N = 6;               % reduction gear ratio
Kt = 5;              % motor torque constant, Nm/amp
Km = Kt;             % motor back EMF constant, V/rad/s
m_res = 1.73;        % motor recistance (ohm)

Jp = 4277.5*4;       % propeller moment of inertia
Jm = 27.4;           % motor moment of inertia
Jdg = 0;
Cm = 0.00022;
Cp = 0;

K1 = Kt/m_res*12;
K2 = Jm + Jp/N^2 +Jdg;
K3 = rho*A*L*gamma;
K4 = rho*A*delta_beta;
%Kh = 17790.0;
K0 = Cm + Cp/N^2 + Kt*Km/m_res;
U0 = 0;

% pitch variables
pitch = 1.5;         % Variable for pitch transfer function
tau_PD = 8;          % Time constant for pitch transfer function.
```

CPP_Healey_TT_2000_Simulink_v3

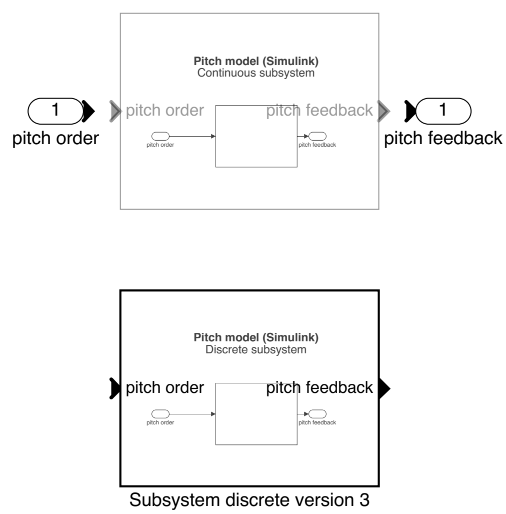


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 1/19

CPP_Healey_TT_2000_Simulink_v3/Subsystem

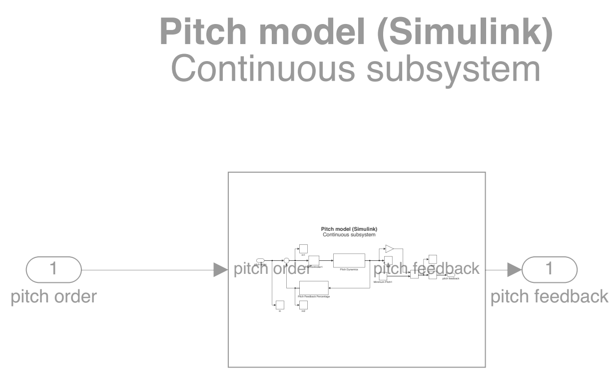
Pitch model (Simulink)
Select subsystem



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 2/19

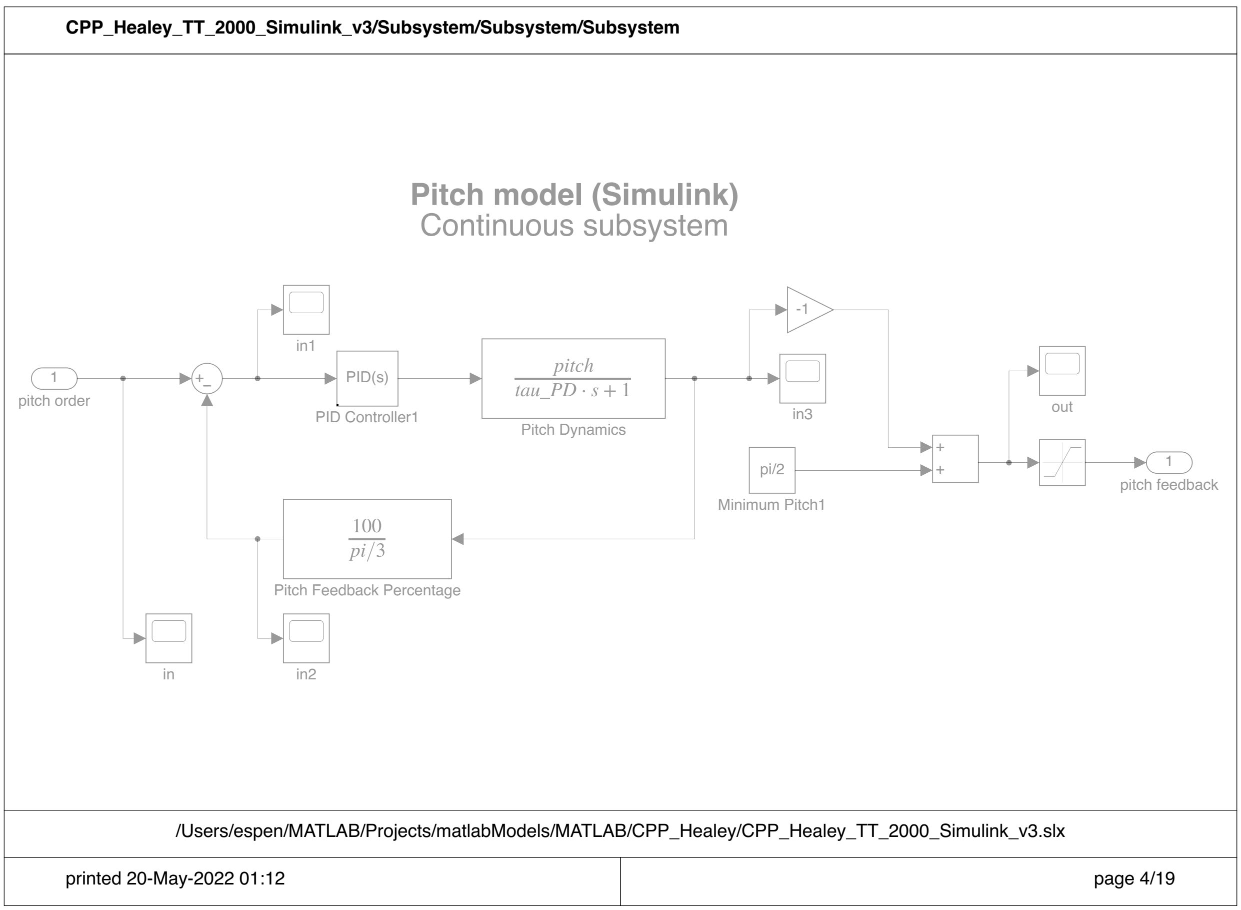
CPP_Healey_TT_2000_Simulink_v3/Subsystem/Subsystem



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

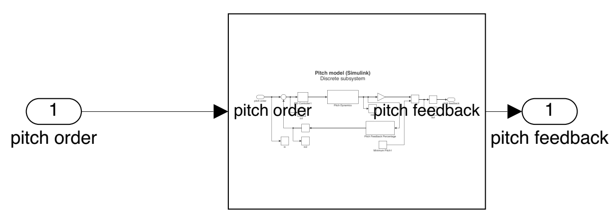
printed 20-May-2022 01:12

page 3/19


CPP_Healey_TT_2000_Simulink_v3/Subsystem/Subsystem/Subsystem
Pitch model (Simulink)
Continuous subsystem
1
pitch order
in1
PID(s)
PID Controller1
pitch
tau_PD · s + 1
Pitch Dynamics
-1
in3
out
pi/2
Minimum Pitch1
1
pitch feedback
100
pi/3
Pitch Feedback Percentage
in
in2
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx
printed 20-May-2022 01:12
page 4/19

CPP_Healey_TT_2000_Simulink_v3/Subsystem/Subsystem discrete version 3

Pitch model (Simulink)
Discrete subsystem

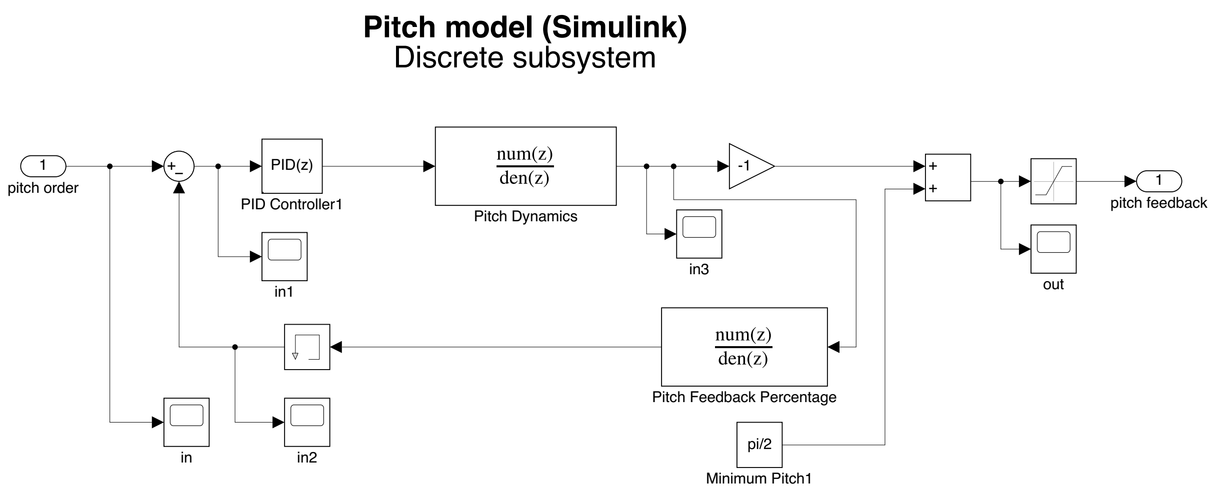


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12

page 5/19

CPP_Healey_TT_2000_Simulink_v3/Subsystem/Subsystem discrete version 3/Subsystem

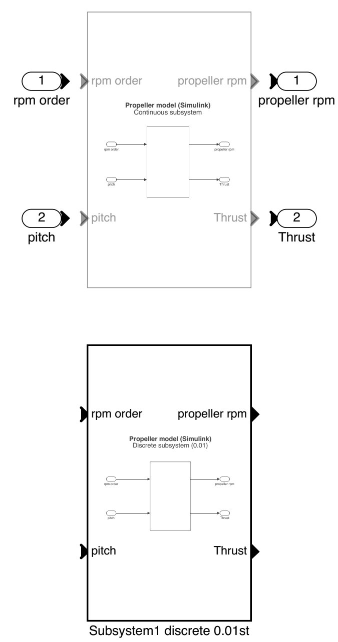


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 6/19

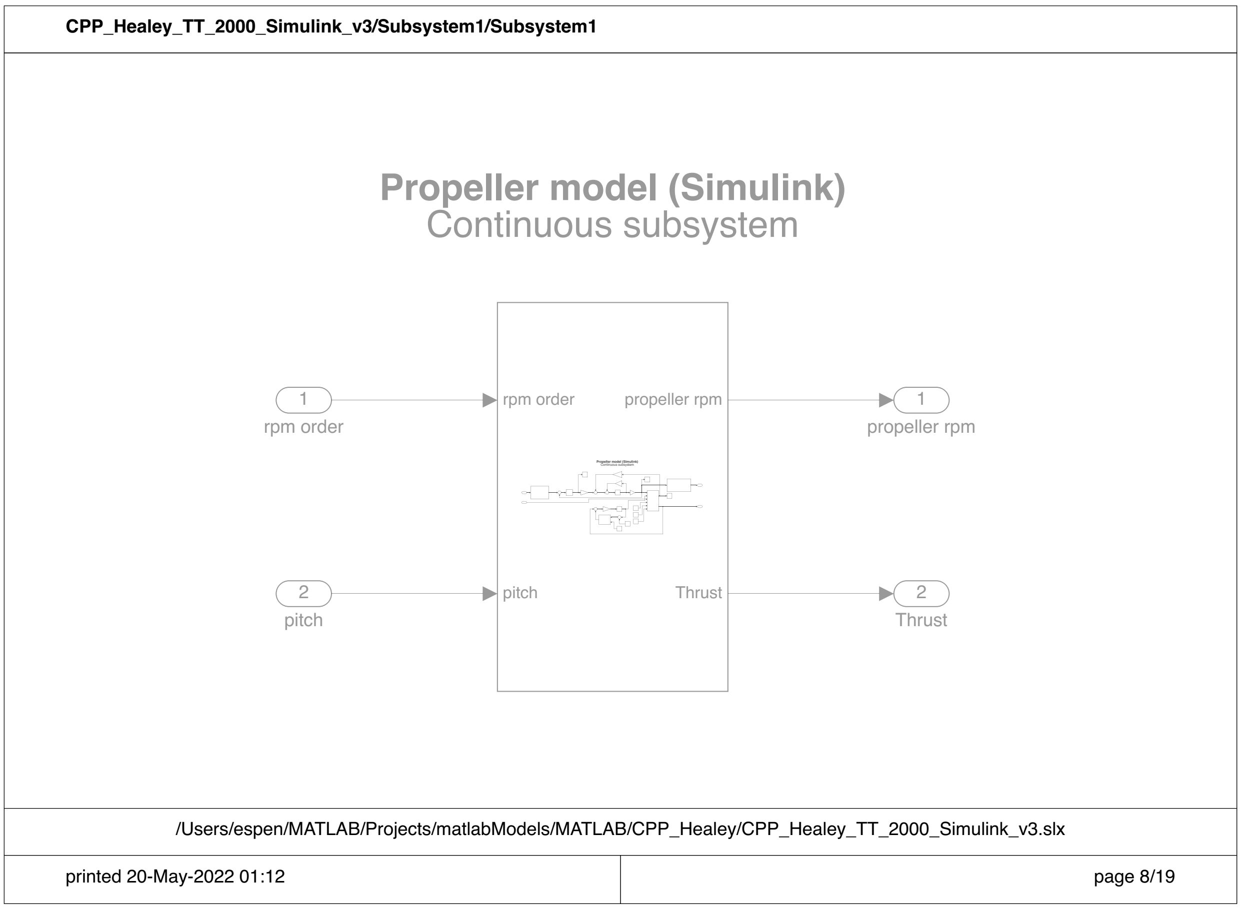
CPP_Healey_TT_2000_Simulink_v3/Subsystem1

Propeller model (Simulink)
Select subsystem

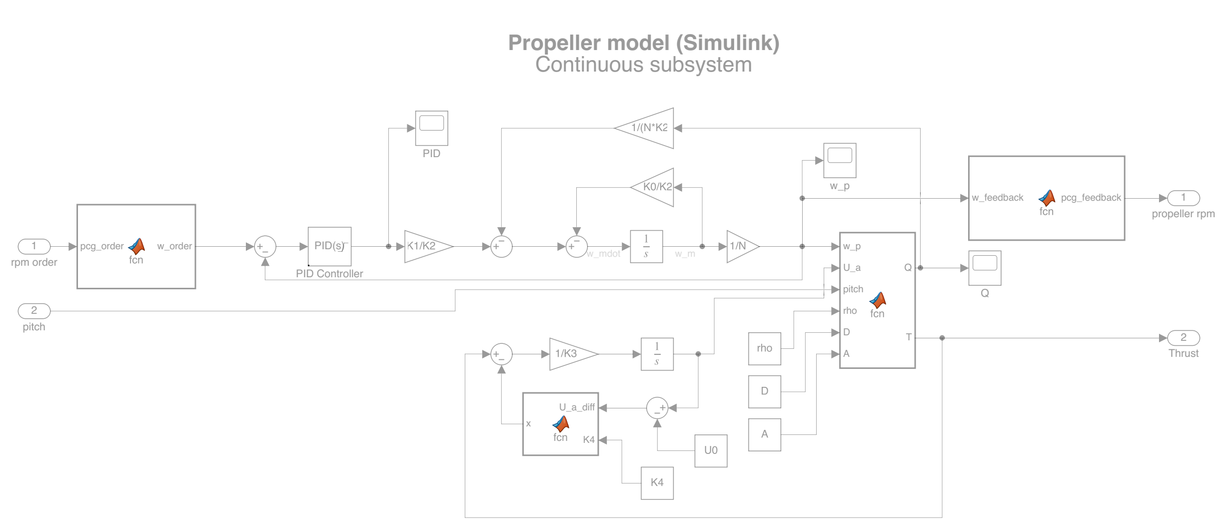


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 7/19


CPP_Healey_TT_2000_Simulink_v3/Subsystem1/Subsystem1
Propeller model (Simulink)
Continuous subsystem
1
rpm order
rpm order
propeller rpm
1
propeller rpm
2
pitch
pitch
Thrust
2
Thrust
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx
printed 20-May-2022 01:12
page 8/19

CPP_Healey_TT_2000_Simulink_v3/Subsystem1/Subsystem1/Subsystem1



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 9/19

```
function [Q,T]= fcn(w_p, U_a, pitch, rho, D, A)

%rho = 998;             % water density
%D = 0.0762;                  % propeller diameter in meter
R = D/2;                % propeller radius in meter
%A = 3.14 * R^2;         % tunnel cross-sectional area
U_p = 0.7 * R * w_p;    % propeller velocity
V_sqr = U_a^2 + U_p^2;  % total relative velocity squared
C_lmax = 1.75;          %
C_dmax = 1.2;           %
% alpha_e = ((pi / 2) - pitch) - atan2(U_a,U_p); % effective angle of attack
% theta = pitch - alpha_e;

theta = atan2(U_a,U_p);
alpha_e = ((pi / 2) - pitch) - theta; % effective angle of attack

Lift = 0.5 * rho * V_sqr * A * C_lmax * sin(2 * alpha_e);
Drag = 0.5 * rho * V_sqr * A * C_dmax * (1 - cos(2 * alpha_e));

Q = 0.7 * R * ((Lift * sin(theta)) + (Drag * cos(theta)));
T = (Lift * cos(theta)) - (Drag * sin(theta));
```

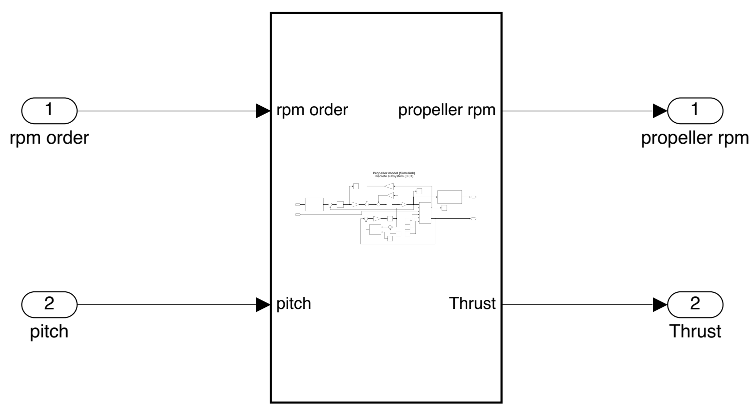
```
function x = fcn(U_a_diff, K4)
x = K4*U_a_diff*abs(U_a_diff);
```

```
function w_order = fcn(pcg_order)
inMin = -100;
inMax = 100;
outMin = -23;
outMax = 23;

w_order = (((pcg_order - inMin) * (outMax - outMin)) / (inMax - inMin)) + outMin;
```

```
function pcg_feedback = fcn(w_feedback)
inMin = -23;
inMax = 23;
outMin = -100;
outMax = 100;

pcg_feedback = (((w_feedback - inMin) * (outMax - outMin)) / (inMax - inMin)) + outMin;
```

CPP_Healey_TT_2000_Simulink_v3/Subsystem1/Subsystem1 discrete 0.01st

Propeller model (Simulink)
Discrete subsystem (0.01)

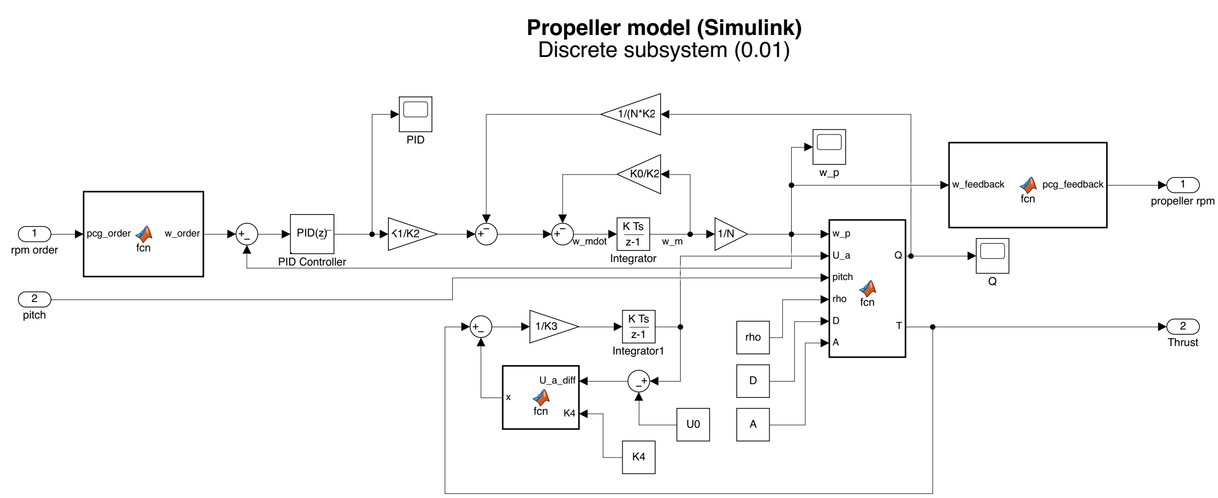


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12

page 14/19

CPP_Healey_TT_2000_Simulink_v3/Subsystem1/Subsystem1 discrete 0.01st/Subsystem1



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/CPP_Healey/CPP_Healey_TT_2000_Simulink_v3.slx

printed 20-May-2022 01:12 | page 15/19

```
function w_order = fcn(pcg_order)
inMin = -100;
inMax = 100;
outMin = -23;
outMax = 23;

w_order = (((pcg_order - inMin) * (outMax - outMin)) / (inMax - inMin)) + outMin;
```

```
function x = fcn(U_a_diff, K4)
x = K4*U_a_diff*abs(U_a_diff);
```

```
function [Q,T]= fcn(w_p, U_a, pitch, rho, D, A)

%rho = 998;             % water density
%D = 0.0762;                  % propeller diameter in meter
R = D/2;                % propeller radius in meter
%A = 3.14 * R^2;         % tunnel cross-sectional area
U_p = 0.7 * R * w_p;    % propeller velocity
V_sqr = U_a^2 + U_p^2;  % total relative velocity squared
C_lmax = 1.75;          %
C_dmax = 1.2;           %
% alpha_e = ((pi / 2) - pitch) - atan2(U_a,U_p); % effective angle of attack
% theta = pitch - alpha_e;

theta = atan2(U_a,U_p);
alpha_e = ((pi / 2) - pitch) - theta; % effective angle of attack

Lift = 0.5 * rho * V_sqr * A * C_lmax * sin(2 * alpha_e);
Drag = 0.5 * rho * V_sqr * A * C_dmax * (1 - cos(2 * alpha_e));

Q = 0.7 * R * ((Lift * sin(theta)) + (Drag * cos(theta)));
T = (Lift * cos(theta)) - (Drag * sin(theta));
```

```
function pcg_feedback = fcn(w_feedback)
inMin = -23;
inMax = 23;
outMin = -100;
outMax = 100;

pcg_feedback = (((w_feedback - inMin) * (outMax - outMin)) / (inMax - inMin)) + outMin;
```

N Azimuth model Matlab/Simulink

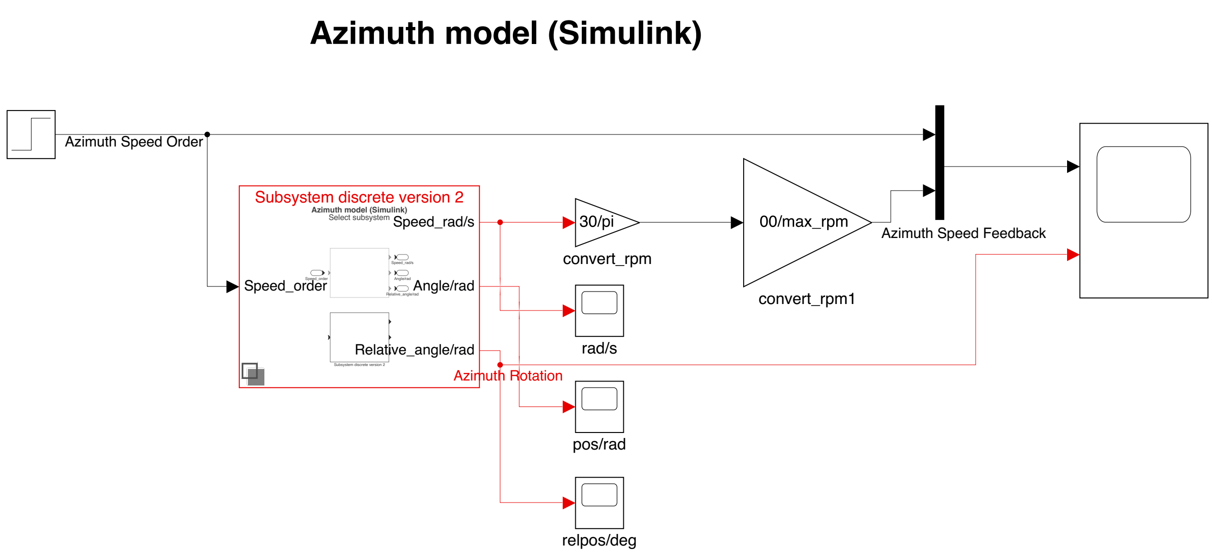
This appendix shows the Matlab program and the Simulink model used for the azimuth model.
Includes:

- Matlab script (1 page)
- Simulink model (8 pages)

Note pages 5 and 8 in the Simulink model describe the function block used in the subsystems.

```
%%
% Azimuth model (MATLAB)
% This program is used to define the parameters used in the Simulink model
%%
clear

M_t = 17000; % The total dryweigth of the thruster

mass_ratio = 0.4;

M_d1 = M_t*mass_ratio;
M_d2 = M_t - M_d1;

R_d1 = 1; % Radius of the propeller
R_d2 = 0.5; % Radius of the thruster shaft

L_d1 = 3; % thruster length

% Calculate the moment of inertia of the lower part of the thruster
I_d1 = M_d1*R_d1^2/4 + M_d1*L_d1^2/12;

% Calculate the moment of inertia of the center part
I_d2 = M_d2*R_d2^2/2;

I_t = I_d1 + I_d2; % The complete moment of inerta of the thruster

I_m = 30; % moment of inerta of the motor


K_t = 0.8;          % motor torque constant, Nm/amp
K_m = K_t;          % motor back EMF constant, V/rad/s
m_res = 0.23;       % motor recistance (ohm)

D_m = 0;            % motor dampning
D_t = 0;            % thruster dampning
max_rpm = 4;        % Max rpm for the azimuth function

K1 = I_m + I_t;
K2 = (D_m + D_t + K_t*K_m/m_res);
K3 = K_t/m_res;

% Create model used for simulation (only used for figures)
model = tf(K3, [K1 K2]);

% Plot the impulse response if enabled
%impulse(model, 100)

% Plot the bode diagram with marginal response if enabled
%figure()
%margin(model)
```

azimuth_model



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx

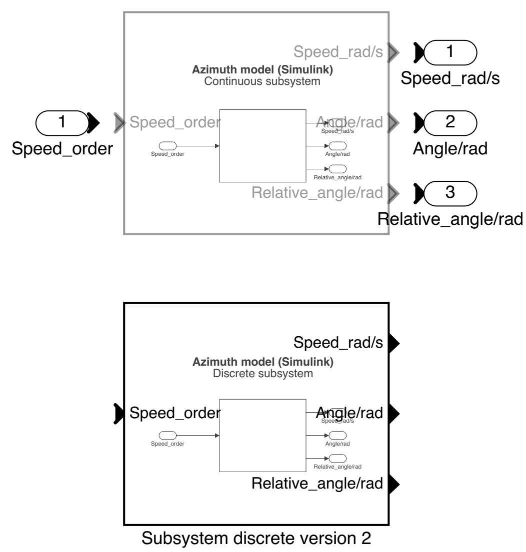
printed 20-May-2022 01:24

page 1/8

azimuth_model/Subsystem

Azimuth model (Simulink)

Select subsystem

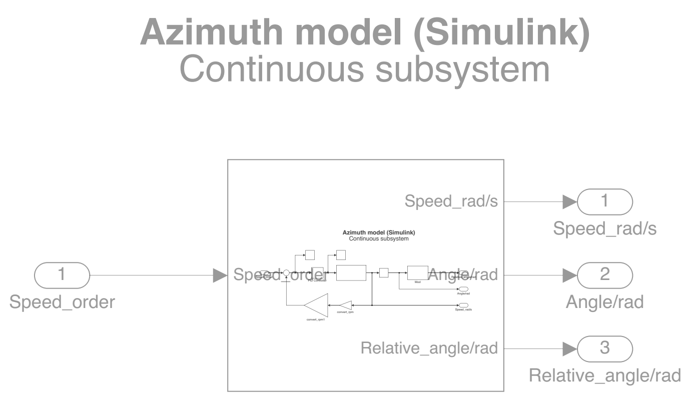


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx

printed 20-May-2022 01:24

page 2/8

azimuth_model/Subsystem/Subsystem

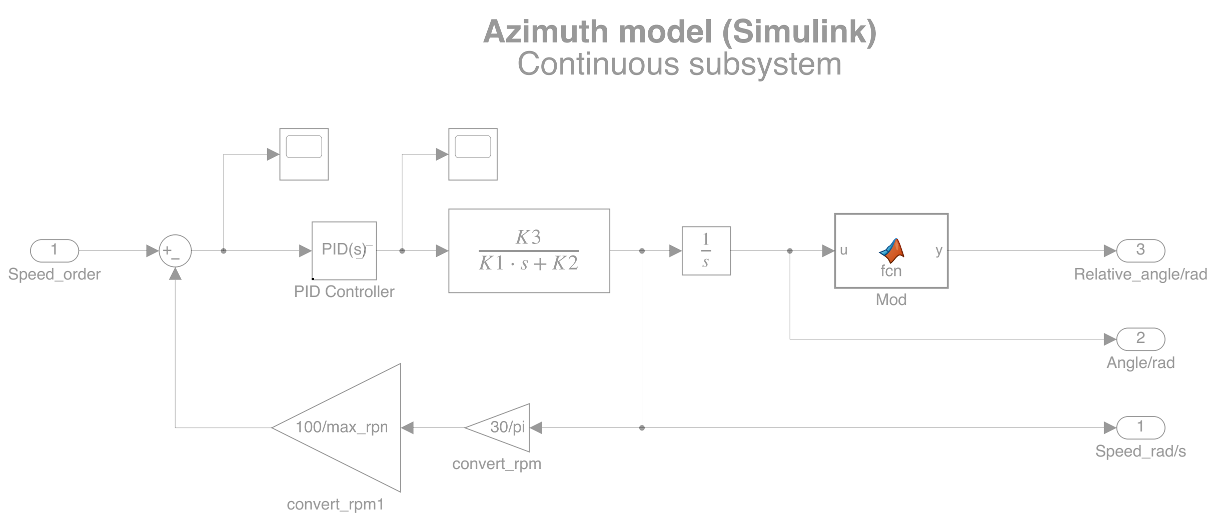


/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx

printed 20-May-2022 01:24

page 3/8

azimuth_model/Subsystem/Subsystem/Subsystem



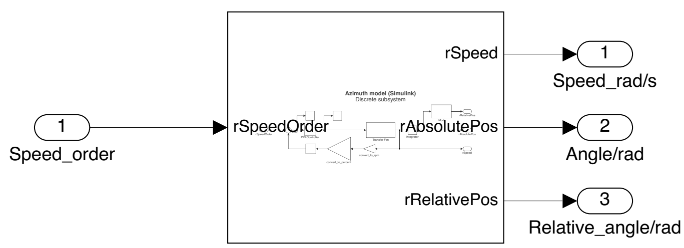
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx

printed 20-May-2022 01:24 | page 4/8

```
function y = fcn(u)
y = mod(u, 2*pi);
```

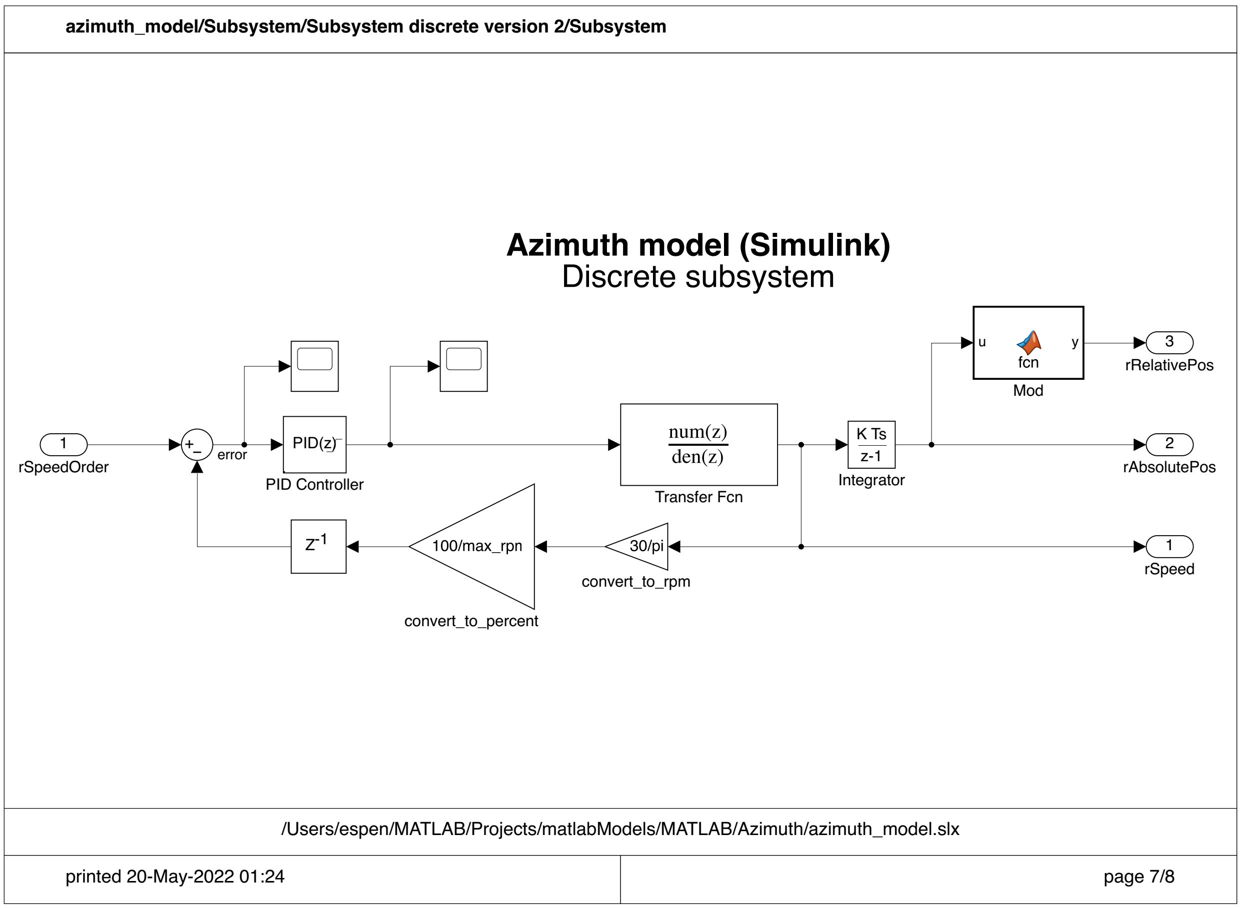
azimuth_model/Subsystem/Subsystem discrete version 2

Azimuth model (Simulink)
Discrete subsystem



/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx

printed 20-May-2022 01:24 page 6/8


azimuth_model/Subsystem/Subsystem discrete version 2/Subsystem
Azimuth model (Simulink)
Discrete subsystem
1
rSpeedOrder
error
PID(z)
PID Controller
num(z)
den(z)
Transfer Fcn
K Ts
z-1
Integrator
u
fcn
y
Mod
3
rRelativePos
2
rAbsolutePos
1
rSpeed
z-1
100/max_rpn
convert_to_percent
30/pi
convert_to_rpm
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/Azimuth/azimuth_model.slx
printed 20-May-2022 01:24
page 7/8

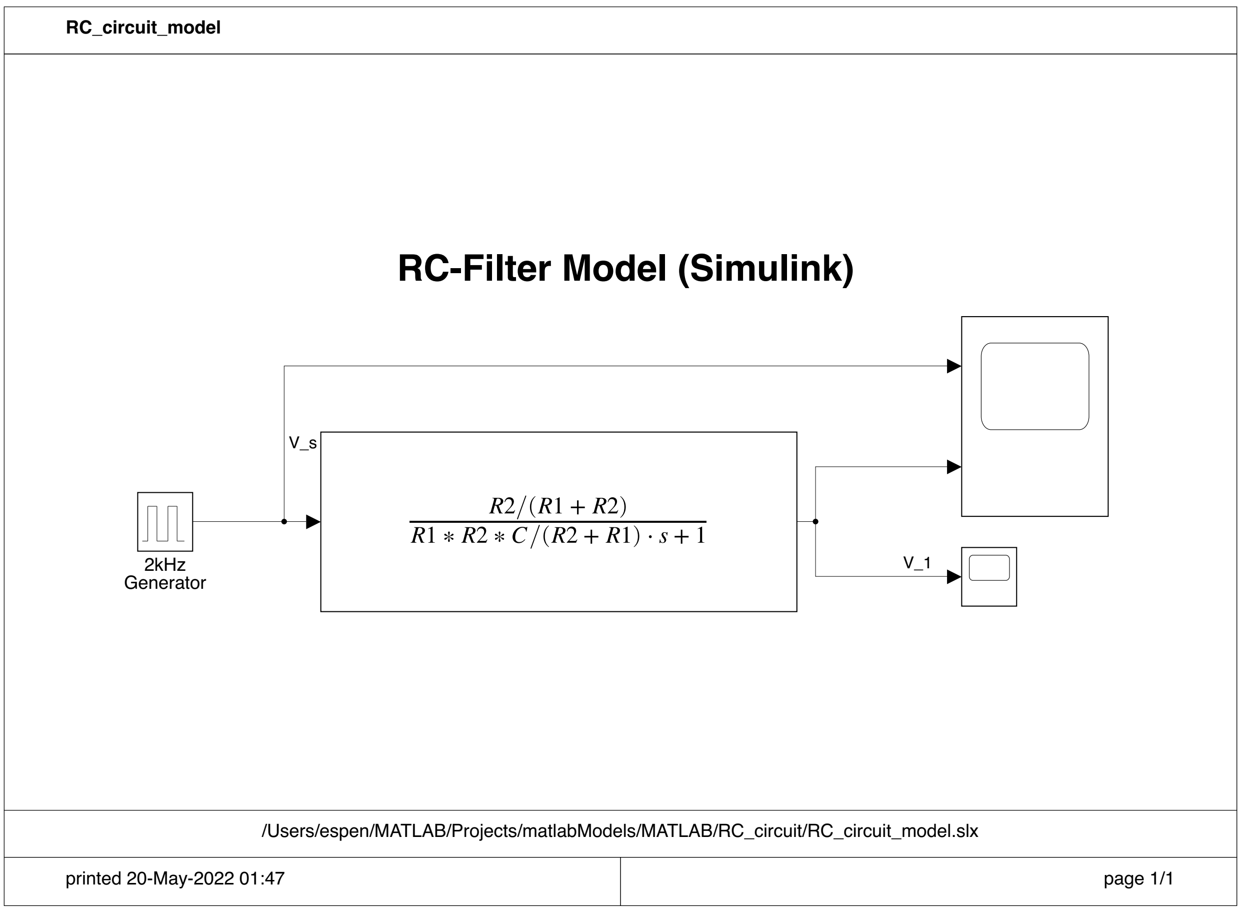
```
function y = fcn(u)
y = mod(u, 2*pi);
```

O RC-Filter model Matlab/Simulink

This appendix shows the Matlab program and the Simulink model used for the RC-Filter model. Includes:

- Matlab script (1 page)
- Simulink model (1 page)

```
%%
% RC-Filter Circuit (MATLAB)
% This is a script to make the TF of the circuit below and model the
% time/phase repsponse.
%
%          R1___
%   Vs >───[___]──────────────> V1
%                 |        |
%               ─────  C  |‾| R2
%               ─────     |_|
%                 |        |
%   V0 >──────────────────────> V0
%
%%

R1 = 14*10^3;       % Set the value of the resistor R1
R2 = 10*10^3;       % Set the value of the resistor R2
C = 2*10^-6;        % Set the value of the capasitor C

% Make the TF
G = tf(R2/(R1+R2), [R2*R1*C/(R2+R1) 1]);

% Plot the bode diagram of the TF with Hz instead of rad/s
plot = bodeplot(G);
setoptions(plot,'FreqUnits','Hz')

% Plot the stepresponse of the circuit
figure()
step(G)

```


RC_circuit_model
RC-Filter Model (Simulink)
V_s
$\frac{R2/(R1+R2)}{R1*R2*C/(R2+R1)\cdot s+1}$
2kHz
Generator
V_1
/Users/espen/MATLAB/Projects/matlabModels/MATLAB/RC_circuit/RC_circuit_model.slx
printed 20-May-2022 01:47
page 1/1

P OPC logger and grapher python

This appendix shows the source code for the Python scripts used for logging OPC UA values and making the graphs from the logs.
Includes:

- OPC logger (3 pages)
- Grapher (2 pages)

Both scripts was adapted for each log/graph.

File - OPC_logger.py

```python
"""
OPC logger
Program used to log OPC UA values to a csv file
"""
import asyncio
import numpy as np
from asyncua import Client
import time


client_address = "opc.tcp://192.168.1.10:4840"

# Handler to interact with subscriptions
class SubHandler:
    """
    Handler to contain different methods to be used when subscribing
    """

    def datachange_notification(self, node, val):
        """
        Called for every datachange notication from the server
        """
    # TODO Add read node ID and print timestamp when variable changed?


def makeWriteableList(time, list1, list2, *list3):
    """Takes in different lists and returns a writeable matrix for use with numpy.savetxt().

    """
    aList = np.array(list1)
    bList = np.array(list2)
    cList = np.array(list3)
    Timer = np.array(time)

    listList = np.transpose(np.array([Timer, aList, bList, *cList]))

    return listList


async def main():
    client = Client(client_address)

    # Connecting to the server
```

File - OPC_logger.py

```
    await client.connect()

    # Assign the desired node to the variable using the NodeId
    # ns = namespace; i = index
    node1 = client.get_node('ns=1;i=100')
    # Getting the value from the object and giving it to the variable
    node1_value = float(await node1.get_value())

    node2 = client.get_node('ns=1;i=101')
    node2_value = float(await node2.get_value())
    node3 = client.get_node('ns=1;i=110')
    node3_value = float(await node2.get_value())
    node4 = client.get_node('ns=1;i=111')
    node4_value = float(await node2.get_value())
    node5 = client.get_node('ns=1;i=112')
    node5_value = float(await node2.get_value())

# ======================    Psuedo subscription ==============================================
    startTime = time.time()
    var1 = []
    var2 = []
    var3 = []
    var4 = []
    var5 = []
    counter = 0
    timer = []
    while counter < 500:

        sub1 = await node1.get_value()
        var1.append(float(sub1))

        sub2 = await node2.get_value()
        var2.append(float(sub2))
        sub3 = await node3.get_value()
        var3.append(float(sub3))
        sub4 = await node4.get_value()
        var4.append(float(sub4))
        sub5 = await node5.get_value()
        var5.append(float(sub5))

        # await asyncio.sleep(0.1)
        counter += 1
        timer.append(time.time()-startTime)

    np.savetxt('log.csv', makeWriteableList(timer, var1,
```

File - OPC_logger.py

```
var2, var3, var4, var5), delimiter=',')

# ======================    End Psuedo subscription
 =============================================

if __name__ == '__main__':
    asyncio.run(main())


```

File - grapher.py

```python
"""
Grapher program
Program to make the graphs used in the thesis
"""
from matplotlib import pyplot as plt
import numpy as np

values = np.loadtxt('log_80rpm_sin20s_100pitch.csv', dtype="float", delimiter=',')

start_time = 0
stop_time = start_time + 100

delta_time = []
last_time = 0
data = []
j = 0

for t in values[:, 0]:
    if start_time <= t < stop_time:
        # Add the delta T
        delta_time.append(t - last_time)
        data.append(values[j])
    last_time = t
    j += 1

# Make array to
delta_time = np.array(delta_time)
data = np.array(data)
time = delta_time.copy()
for i in np.arange(delta_time.size):
    time[i] = np.sum(delta_time[0:i])

#Enable new figure
#fig = plt.figure(dpi=200, figsize=(12.8, 7), num=1)
fig = plt.figure(dpi=200, figsize=(9, 7), num=1)

top_fig = fig.add_subplot(211)
#Plotting x and y values and then addidtonal style options
top_fig.plot(time[:], data[:, 2], linestyle='dashed', label='RPM Order')
top_fig.plot(time, data[:, 1], label='RPM Feedback')
top_fig.plot(time, data[:, 4], linestyle='dashed', label='Pitch Order')
top_fig.plot(time, data[:, 3], label='Pitch Feedback')
# ax1.plot(values[:, 0], values[:, 5], label='Thrust')

bottom_fig = fig.add_subplot(212, sharex=top_fig)
```

File - grapher.py

```
#bottom_fig = top_fig.twinx()
bottom_fig.set_ylabel('Y2-axis')
bottom_fig.plot(time, data[:, 5]/1000, label='Thrust')

bottom_fig.set_ylim([-300, 300])
#Customizing plot
#Putting labels in X and Y axis
bottom_fig.set_xlabel('Time [s]')
top_fig.set_ylabel('Values [%]')
bottom_fig.set_ylabel('Output Thrust [kN]')

top_fig.margins(0, 0.1)
top_fig.grid(which='both', axis='both')  # Enable the grid
bottom_fig.margins(0, 0.1)
bottom_fig.grid(which='both', axis='both')  # Enable the grid
#fig.tight_layout()

bottom_fig.legend()
top_fig.legend()
top_fig.set_title('RPM and Pitch')
bottom_fig.set_title('Thrust')

fig.savefig("logtest")
#Showing figure
plt.show()

```

Q Workflow script

This appendix shows the source code for the script used to automate the 4diac development.

Includes:

- Automate workflow script (3 pages)

File - automate_workflow_script.sh

```
#! /bin/bash
##
# Workflow automation script
# This program is used to automate the building process of Forte on Linux
##
function getRemote() {
  # Set the default ip address
  ADDRESS=root@192.168.1.10

  echo -en "Specify the shh hostname (root@192.168.1.10): "
  read new_ip

  # If the new_ip matches a correct IP address it is used
  if [[ ${new_ip} =~ [0-z]*\@[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3} ]]
  then
    ADDRESS=$new_ip
    echo -e "Using the user specified: ${ADDRESS}"
  else
    echo -e "Using the default: ${ADDRESS}"
  fi
}

while true
do
echo "What do you want to do?"
echo "1: Make local version of FORTE"
echo "2: Make remote version of FORTE (PFC200)"
echo "3: Send bootfiles to remote (predefined paths and names)"
echo "0: Exit"
echo -en "Choice: "
read choice


case $choice in
    1)
      echo -e "Making local verison of FORTE"
      cd ${HOME}/FORTE_dev/bin/posix/

      cmake -DCMAKE_BUILD_TYPE:STRING=Debug \
      -DFORTE_COM_ETH:BOOL=ON -DFORTE_COM_FBDK:BOOL=ON \
      -DFORTE_COM_LOCAL:BOOL=ON -DFORTE_COM_RAW:BOOL=ON \
      -DFORTE_ARCHITECTURE:STRING=Posix -DFORTE_MODULE_CONVERT:BOOL=ON \
      -DFORTE_TRACE_EVENTS:BOOL=OFF -DFORTE_MODULE_IEC61131=ON \
```

File - automate_workflow_script.sh

```
      -DFORTE_EXTERNAL_MODULES_DIRECTORY=${HOME}/FORTE_dev/ext_modules \
      -DFORTE_MODULE_EXTERNAL_mconLib=ON -DFORTE_COM_OPC_UA=ON \
      -DFORTE_COM_OPC_UA_INCLUDE_DIR=${HOME}/FORTE_dev/open62541/build \
      -DFORTE_COM_OPC_UA_LIB=libopen62541.so \
      -DFORTE_COM_OPC_UA_LIB_DIR=${HOME}/FORTE_dev/open62541/build/bin \
      -DFORTE_COM_OPC_UA_PORT=4840 -DFORTE_COM_OPC_UA_CLIENT_PUB_INTERVAL=100.0 \
      -S ${HOME}/FORTE_dev/forte_src/ -B ${HOME}/FORTE_dev/bin/posix/

      make
    ;;

    2)
      echo -e "Making remote verison of FORTE"

      # Make the new verison using ptxdist
      cd ${HOME}/wago/ptxproj
      ptxdist clean forte_wago
      ptxdist install forte_wago

      echo -en "Send to remote? [y/n]"
      read remote

      if [[ ${remote} =~ (yes)|y ]]
      then
        getRemote

        # Send the new version to a device:
        cd ${HOME}/wago/ptxproj/platform-wago-pfcXXX/build-target/forte_wago-1.6.2-build/src
        scp forte ${ADDRESS}:/usr/bin/
      fi
    ;;

    3)
      # Send to ip-address:
        echo -e "Sending bootfiles to remote"
  getRemote

        # Send files to the remote


        scp -r $HOME/FORTE_dev/forte-boot/ ${ADDRESS}:/home
```

File - automate_workflow_script.sh

```
/admin/

        # Go to the romote and move the file to the docker volume
        # restart the docker container
        #ssh pi@10.0.0.237:/home/admin/forte-boot/ 'docker cp thrusterSimulator_A200_OPC.fboot docker_forte_OPC_1:/app/ | docker-compose restart'

    ;;

    0)
      echo -e "EXIT"
      exit
    ;;
esac
done
```

R Log raw data

Due to the size of the logs there is only included two logs. The logs for creating Figures **??** and 5.16a. Includes:

- Step response of the new propeller model (15 pages)
- Step response of the azimuth function (15 pages)

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
2.491948604583740234e-01,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
4.996500015258789062e-01,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
7.501847743988037109e-01,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
1.000364065170288086e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
1.249814987182617188e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
1.500747919082641602e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
1.748261928558349609e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
2.000401735305786133e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
2.249118804931640625e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
2.500005722045898438e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
2.750138759613037109e+00,0.000000000000000000e+00,0.000000000000000000e+00,-2.828979492187500000e-02,0.
000000000000000000e+00,-0.000000000000000000e+00
3.000830888748168945e+00,0.000000000000000000e+00,0.000000000000000000e+00,2.552841186523437500e+00,9.
000000000000000000e+01,1.987354248046875000e+03
3.249934911727905273e+00,6.741707324981689453e+00,1.000000000000000000e+02,1.241921997070312500e+01,9.
000000000000000000e+01,1.145383593750000000e+04
3.501791000366210938e+00,2.905628395080566406e+01,1.000000000000000000e+02,2.498271942138671875e+01,9.
000000000000000000e+01,2.479228320312500000e+04
3.749529123306274414e+00,4.370970916748046875e+01,1.000000000000000000e+02,3.204332733154296875e+01,9.
000000000000000000e+01,3.133636523437500000e+04
4.000431776046752930e+00,6.479570007324218750e+01,1.000000000000000000e+02,4.103556060791015625e+01,9.
000000000000000000e+01,2.963423242187500000e+04
4.248735904693603516e+00,7.154419708251953125e+01,1.000000000000000000e+02,4.614106750488281250e+01,9.
000000000000000000e+01,3.399579687500000000e+04
4.500139951705932617e+00,7.572028350830078125e+01,1.000000000000000000e+02,5.271075439453125000e+01,9.
000000000000000000e+01,4.337546875000000000e+04
4.748720884323120117e+00,7.758244323730468750e+01,1.000000000000000000e+02,5.648088836669921875e+01,9.
000000000000000000e+01,5.040516015625000000e+04
5.000323057174682617e+00,8.002767944335937500e+01,1.000000000000000000e+02,6.138343048095703125e+01,9.
000000000000000000e+01,6.153587109375000000e+04
5.250661849975585938e+00,8.153177642822265625e+01,1.000000000000000000e+02,6.422679138183593750e+01,9.
000000000000000000e+01,6.918740625000000000e+04
5.500025033950805664e+00,8.363258361816406250e+01,1.000000000000000000e+02,6.796175384521484375e+01,9.
000000000000000000e+01,8.081207031250000000e+04
5.751031875610351562e+00,8.493244934082031250e+01,1.000000000000000000e+02,7.014953613281250000e+01,9.
000000000000000000e+01,8.856362500000000000e+04
6.000340938568115234e+00,8.673300170898437500e+01,1.000000000000000000e+02,7.304987335205078125e+01,9.
000000000000000000e+01,1.000526718750000000e+05
6.249847888946533203e+00,8.783541107177734375e+01,1.000000000000000000e+02,7.476369476318359375e+01,9.
000000000000000000e+01,1.075506484375000000e+05
6.500491857528686523e+00,8.934628295898437500e+01,1.000000000000000000e+02,7.705361175537109375e+01,9.
000000000000000000e+01,1.184575625000000000e+05
6.750276088714599609e+00,9.026160430908203125e+01,1.000000000000000000e+02,7.841656494140625000e+01,9.
000000000000000000e+01,1.254562656250000000e+05
7.000653028488159180e+00,9.150333404541015625e+01,1.000000000000000000e+02,8.024907684326171875e+01,9.
000000000000000000e+01,1.354846250000000000e+05
7.250349044799804688e+00,9.224801635742187500e+01,1.000000000000000000e+02,8.134584808349609375e+01,9.
000000000000000000e+01,1.418307812500000000e+05
7.500673055648803711e+00,9.324838256835937500e+01,1.000000000000000000e+02,8.282719421386718750e+01,9.
000000000000000000e+01,1.508100937500000000e+05
7.748502016067504883e+00,9.384245300292968750e+01,1.000000000000000000e+02,8.371715545654296875e+01,9.
000000000000000000e+01,1.564256562500000000e+05
8.000836849212646484e+00,9.463290405273437500e+01,1.000000000000000000e+02,8.492258453369140625e+01,9.
000000000000000000e+01,1.642852968750000000e+05
8.150124073028564453e+00,9.509782409667968750e+01,1.000000000000000000e+02,8.529279327392578125e+01,9.
000000000000000000e+01,1.667549687500000000e+05
8.400464773178100586e+00,9.551709747314453125e+01,1.000000000000000000e+02,8.631756591796875000e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,1.737186562500000000e+05
8.648453712463378906e+00,9.589410400390625000e+01,1.000000000000000000e+02,8.693481445312500000e+01,9.
000000000000000000e+01,1.779968281250000000e+05
8.900729894638061523e+00,9.638748931884765625e+01,1.000000000000000000e+02,8.777171325683593750e+01,9.
000000000000000000e+01,1.838856875000000000e+05
9.148660898208618164e+00,9.667292022705078125e+01,1.000000000000000000e+02,8.827539062500000000e+01,9.
000000000000000000e+01,1.874724687500000000e+05
9.401244878768920898e+00,9.704328155517578125e+01,1.000000000000000000e+02,8.895719909667968750e+01,9.
000000000000000000e+01,1.923695625000000000e+05
9.650438785552978516e+00,9.725578308105468750e+01,1.000000000000000000e+02,8.936654663085937500e+01,9.
000000000000000000e+01,1.953284375000000000e+05
9.900086879730224609e+00,9.752950286865234375e+01,1.000000000000000000e+02,8.991888427734375000e+01,9.
000000000000000000e+01,1.993371718750000000e+05
1.015060877799987793e+01,9.768553924560546875e+01,1.000000000000000000e+02,9.024916076660156250e+01,9.
000000000000000000e+01,2.017408125000000000e+05
1.040002775192260742e+01,9.788553619384765625e+01,1.000000000000000000e+02,9.069259643554687500e+01,9.
000000000000000000e+01,2.049730156250000000e+05
1.065043497085571289e+01,9.799919128417968750e+01,1.000000000000000000e+02,9.095616149902343750e+01,9.
000000000000000000e+01,2.068964375000000000e+05
1.090188789367675781e+01,9.814484405517578125e+01,1.000000000000000000e+02,9.130752563476562500e+01,9.
000000000000000000e+01,2.094635781250000000e+05
1.115354800224304199e+01,9.822787475585937500e+01,1.000000000000000000e+02,9.151461791992187500e+01,9.
000000000000000000e+01,2.109794687500000000e+05
1.140075278282165527e+01,9.833512115478515625e+01,1.000000000000000000e+02,9.178798675537109375e+01,9.
000000000000000000e+01,2.129869531250000000e+05
1.165025877952575684e+01,9.839707946777343750e+01,1.000000000000000000e+02,9.194721984863281250e+01,9.
000000000000000000e+01,2.141626250000000000e+05
1.190015292167663574e+01,9.847871398925781250e+01,1.000000000000000000e+02,9.215449523925781250e+01,9.
000000000000000000e+01,2.157063437500000000e+05
1.215137410163879395e+01,9.852716064453125000e+01,1.000000000000000000e+02,9.227320861816406250e+01,9.
000000000000000000e+01,2.166021250000000000e+05
1.239980006217956543e+01,9.859317779541015625e+01,1.000000000000000000e+02,9.242458343505859375e+01,9.
000000000000000000e+01,2.177667968750000000e+05
1.265065193176269531e+01,9.863392639160156250e+01,1.000000000000000000e+02,9.250907135009765625e+01,9.
000000000000000000e+01,2.184353437500000000e+05
1.290141105651855469e+01,9.869187927246093750e+01,1.000000000000000000e+02,9.261332702636718750e+01,9.
000000000000000000e+01,2.192940937500000000e+05
1.315076088905334473e+01,9.872924804687500000e+01,1.000000000000000000e+02,9.266904449462890625e+01,9.
000000000000000000e+01,2.197802968750000000e+05
1.340136694908142090e+01,9.878462982177734375e+01,1.000000000000000000e+02,9.273380279541015625e+01,9.
000000000000000000e+01,2.203950000000000000e+05
1.364949488639831543e+01,9.882170104980468750e+01,1.000000000000000000e+02,9.276547241210937500e+01,9.
000000000000000000e+01,2.207365000000000000e+05
1.390157389640808105e+01,9.887831115722656250e+01,1.000000000000000000e+02,9.279734039306640625e+01,9.
000000000000000000e+01,2.211585937500000000e+05
1.414821696281433105e+01,9.891711425781250000e+01,1.000000000000000000e+02,9.280906677246093750e+01,9.
000000000000000000e+01,2.213865000000000000e+05
1.440041804313659668e+01,9.897737884521484375e+01,1.000000000000000000e+02,9.281378173828125000e+01,9.
000000000000000000e+01,2.216580937500000000e+05
1.465144181251525879e+01,9.901913452148437500e+01,1.000000000000000000e+02,9.280913543701171875e+01,9.
000000000000000000e+01,2.217976718750000000e+05
1.490067791938781738e+01,9.908437347412109375e+01,1.000000000000000000e+02,9.279168701171875000e+01,9.
000000000000000000e+01,2.219528125000000000e+05
1.515107679367065430e+01,9.912966156005859375e+01,1.000000000000000000e+02,9.277375793457031250e+01,9.
000000000000000000e+01,2.220242968750000000e+05
1.540088295936584473e+01,9.920033264160156250e+01,1.000000000000000000e+02,9.273846435546875000e+01,9.
000000000000000000e+01,2.220900937500000000e+05
1.565049076080322266e+01,9.924923706054687500e+01,1.000000000000000000e+02,9.270993041992187500e+01,9.
000000000000000000e+01,2.221096093750000000e+05
1.590130305290222168e+01,9.932517242431640625e+01,1.000000000000000000e+02,9.266053009033203125e+01,9.
000000000000000000e+01,2.221073281250000000e+05
1.615006780624389648e+01,9.937741851806640625e+01,1.000000000000000000e+02,9.262370300292968750e+01,9.
000000000000000000e+01,2.220873750000000000e+05
1.640089273452758789e+01,9.945799255371093750e+01,1.000000000000000000e+02,9.256339263916015625e+01,9.
000000000000000000e+01,2.220335937500000000e+05
1.664924693107604980e+01,9.951303100585937500e+01,1.000000000000000000e+02,9.252024841308593750e+01,9.
000000000000000000e+01,2.219838437500000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
1.690101194381713867e+01,9.959729003906250000e+01,1.000000000000000000e+02,9.245178222656250000e+01,9.
000000000000000000e+01,2.218912187500000000e+05
1.715077877044677734e+01,9.965441894531250000e+01,1.000000000000000000e+02,9.240402221679687500e+01,9.
000000000000000000e+01,2.218189531250000000e+05
1.740062093734741211e+01,9.974121856689453125e+01,1.000000000000000000e+02,9.232975006103515625e+01,9.
000000000000000000e+01,2.216970468750000000e+05
1.765136194229125977e+01,9.979962921142578125e+01,1.000000000000000000e+02,9.227880096435546875e+01,9.
000000000000000000e+01,2.216077656250000000e+05
1.790203690528869629e+01,9.988771820068359375e+01,1.000000000000000000e+02,9.220070648193359375e+01,9.
000000000000000000e+01,2.214636093750000000e+05
1.810072588920593262e+01,9.994655609130859375e+01,1.000000000000000000e+02,9.214778137207031250e+01,9.
000000000000000000e+01,2.213614218750000000e+05
1.835026884078979492e+01,1.000053329467773438e+02,1.000000000000000000e+02,9.209436798095703125e+01,9.
000000000000000000e+01,2.212548750000000000e+05
1.860133194923400879e+01,1.000930862426757812e+02,1.000000000000000000e+02,9.201364898681640625e+01,9.
000000000000000000e+01,2.210878750000000000e+05
1.885225796699523926e+01,1.001511154174804688e+02,1.000000000000000000e+02,9.195963287353515625e+01,9.
000000000000000000e+01,2.209722500000000000e+05
1.910051608085632324e+01,1.002371520996093750e+02,1.000000000000000000e+02,9.187861633300781250e+01,9.
000000000000000000e+01,2.207929687500000000e+05
1.935042691230773926e+01,1.002936477661132812e+02,1.000000000000000000e+02,9.182477569580078125e+01,9.
000000000000000000e+01,2.206699531250000000e+05
1.960106277465820312e+01,1.003768310546875000e+02,1.000000000000000000e+02,9.174452209472656250e+01,9.
000000000000000000e+01,2.204806250000000000e+05
1.985154390335083008e+01,1.004310836791992188e+02,1.000000000000000000e+02,9.169148254394531250e+01,9.
000000000000000000e+01,2.203514687500000000e+05
2.010144591331481934e+01,1.005103912353515625e+02,1.000000000000000000e+02,9.161283111572265625e+01,9.
000000000000000000e+01,2.201535781250000000e+05
2.035075783729553223e+01,1.005617523193359375e+02,1.000000000000000000e+02,9.156110382080078125e+01,9.
000000000000000000e+01,2.200191250000000000e+05
2.060158896446228027e+01,1.006363067626953125e+02,1.000000000000000000e+02,9.148473358154296875e+01,9.
000000000000000000e+01,2.198137500000000000e+05
2.085051894187927246e+01,1.006842346191406250e+02,1.000000000000000000e+02,9.143472290039062500e+01,9.
000000000000000000e+01,2.196744843750000000e+05
2.110095477104187012e+01,1.007532806396484375e+02,1.000000000000000000e+02,9.136114501953125000e+01,9.
000000000000000000e+01,2.195334531250000000e+05
2.130042481422424316e+01,1.007973251342773438e+02,1.000000000000000000e+02,9.131314086914062500e+01,9.
000000000000000000e+01,2.193186875000000000e+05
2.154993581771850586e+01,1.008397140502929688e+02,1.000000000000000000e+02,9.126600646972656250e+01,9.
000000000000000000e+01,2.191733750000000000e+05
2.180073285102844238e+01,1.009000854492187500e+02,1.000000000000000000e+02,9.119699859619140625e+01,9.
000000000000000000e+01,2.189524375000000000e+05
2.205202507972717285e+01,1.009381256103515625e+02,1.000000000000000000e+02,9.115216064453125000e+01,9.
000000000000000000e+01,2.188031875000000000e+05
2.230170392990112305e+01,1.009917907714843750e+02,1.000000000000000000e+02,9.108671569824218750e+01,9.
000000000000000000e+01,2.185764375000000000e+05
2.255069279670715332e+01,1.010252685546875000e+02,1.000000000000000000e+02,9.104431152343750000e+01,9.
000000000000000000e+01,2.184234531250000000e+05
2.280176472663879395e+01,1.010719757080078125e+02,1.000000000000000000e+02,9.098258209228515625e+01,9.
000000000000000000e+01,2.181912968750000000e+05
2.304919886589050293e+01,1.011007614135742188e+02,1.000000000000000000e+02,9.094268798828125000e+01,9.
000000000000000000e+01,2.180348125000000000e+05
2.330118799209594727e+01,1.011403732299804688e+02,1.000000000000000000e+02,9.088475036621093750e+01,9.
000000000000000000e+01,2.177976093750000000e+05
2.355145907402038574e+01,1.011644134521484375e+02,1.000000000000000000e+02,9.084739685058593750e+01,9.
000000000000000000e+01,2.176379375000000000e+05
2.380135297775268555e+01,1.011969223022460938e+02,1.000000000000000000e+02,9.079327392578125000e+01,9.
000000000000000000e+01,2.173962031250000000e+05
2.405174279212951660e+01,1.012162399291992188e+02,1.000000000000000000e+02,9.075845336914062500e+01,9.
000000000000000000e+01,2.172336250000000000e+05
2.430072283744812012e+01,1.012417297363281250e+02,1.000000000000000000e+02,9.070811462402343750e+01,9.
000000000000000000e+01,2.169878593750000000e+05
2.455317091941833496e+01,1.012564239501953125e+02,1.000000000000000000e+02,9.067579650878906250e+01,9.
000000000000000000e+01,2.168228125000000000e+05
2.480121493339538574e+01,1.012750854492187500e+02,1.000000000000000000e+02,9.062916564941406250e+01,9.
000000000000000000e+01,2.165736562500000000e+05
2.505149698257446289e+01,1.012853088378906250e+02,1.000000000000000000e+02,9.059928894042968750e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,2.164065625000000000e+05
2.530221986770629883e+01,1.012973937988281250e+02,1.000000000000000000e+02,9.055625152587890625e+01,9.
000000000000000000e+01,2.161547031250000000e+05
2.555046796798706055e+01,1.013033370971679688e+02,1.000000000000000000e+02,9.052873229980468750e+01,9.
000000000000000000e+01,2.159860468750000000e+05
2.580113101005554199e+01,1.013091888427734375e+02,1.000000000000000000e+02,9.048917388916015625e+01,9.
000000000000000000e+01,2.157322031250000000e+05
2.605160498619079590e+01,1.013111038208007812e+02,1.000000000000000000e+02,9.046392059326171875e+01,9.
000000000000000000e+01,2.155625312500000000e+05
2.629860281944274902e+01,1.013110961914062500e+02,1.000000000000000000e+02,9.042768859863281250e+01,9.
000000000000000000e+01,2.153925468750000000e+05
2.650254487991333008e+01,1.013092422485351562e+02,1.000000000000000000e+02,9.040460205078125000e+01,9.
000000000000000000e+01,2.151373437500000000e+05
2.675065803527832031e+01,1.013059692382812500e+02,1.000000000000000000e+02,9.038234710693359375e+01,9.
000000000000000000e+01,2.149671406250000000e+05
2.700192689895629883e+01,1.012985000610351562e+02,1.000000000000000000e+02,9.035048675537109375e+01,9.
000000000000000000e+01,2.147120312500000000e+05
2.725212597846984863e+01,1.012918853759765625e+02,1.000000000000000000e+02,9.033024597167968750e+01,9.
000000000000000000e+01,2.145421562500000000e+05
2.750138688087463379e+01,1.012796401977539062e+02,1.000000000000000000e+02,9.030130004882812500e+01,9.
000000000000000000e+01,2.142880312500000000e+05
2.775140404701232910e+01,1.012700042724609375e+02,1.000000000000000000e+02,9.028294372558593750e+01,9.
000000000000000000e+01,2.141191406250000000e+05
2.800122380256652832e+01,1.012534637451171875e+02,1.000000000000000000e+02,9.025674438476562500e+01,9.
000000000000000000e+01,2.138667968750000000e+05
2.814969182014465332e+01,1.012411346435546875e+02,1.000000000000000000e+02,9.024835968017578125e+01,9.
000000000000000000e+01,2.137830468750000000e+05
2.840118384361267090e+01,1.012278213500976562e+02,1.000000000000000000e+02,9.022422790527343750e+01,9.
000000000000000000e+01,2.135329218750000000e+05
2.864990282058715820e+01,1.012135772705078125e+02,1.000000000000000000e+02,9.020896148681640625e+01,9.
000000000000000000e+01,2.133672343750000000e+05
2.890108799934387207e+01,1.011905670166015625e+02,1.000000000000000000e+02,9.018724822998046875e+01,9.
000000000000000000e+01,2.131205468750000000e+05
2.915098381042480469e+01,1.011742248535156250e+02,1.000000000000000000e+02,9.017353820800781250e+01,9.
000000000000000000e+01,2.129574531250000000e+05
2.940158581733703613e+01,1.011483154296875000e+02,1.000000000000000000e+02,9.015406799316406250e+01,9.
000000000000000000e+01,2.127151250000000000e+05
2.964958882331848145e+01,1.011301879882812500e+02,1.000000000000000000e+02,9.014178466796875000e+01,9.
000000000000000000e+01,2.125552031250000000e+05
2.990225672721862793e+01,1.011018524169921875e+02,1.000000000000000000e+02,9.012438964843750000e+01,9.
000000000000000000e+01,2.123179843750000000e+05
3.015162587165832520e+01,1.010822677612304688e+02,1.000000000000000000e+02,9.011344909667968750e+01,9.
000000000000000000e+01,2.121617343750000000e+05
3.040193390846252441e+01,1.010519638061523438e+02,1.000000000000000000e+02,9.009795379638671875e+01,9.
000000000000000000e+01,2.119304062500000000e+05
3.065092611312866211e+01,1.010312194824218750e+02,1.000000000000000000e+02,9.008822631835937500e+01,9.
000000000000000000e+01,2.117782968750000000e+05
3.090256690979003906e+01,1.009993896484375000e+02,1.000000000000000000e+02,9.007447814941406250e+01,9.
000000000000000000e+01,2.115535468750000000e+05
3.115166211128234863e+01,1.009777603149414062e+02,1.000000000000000000e+02,9.006586456298828125e+01,9.
000000000000000000e+01,2.114060625000000000e+05
3.140124106407165527e+01,1.009448165893554688e+02,1.000000000000000000e+02,9.005371856689453125e+01,9.
000000000000000000e+01,2.111885312500000000e+05
3.165073299407958984e+01,1.009225845336914062e+02,1.000000000000000000e+02,9.004611968994140625e+01,9.
000000000000000000e+01,2.110460312500000000e+05
3.190102291107177734e+01,1.008889083862304688e+02,1.000000000000000000e+02,9.003543090820312500e+01,9.
000000000000000000e+01,2.108362812500000000e+05
3.215193080902099609e+01,1.008663101196289062e+02,1.000000000000000000e+02,9.002875518798828125e+01,9.
000000000000000000e+01,2.106991093750000000e+05
3.240190386772155762e+01,1.008322753906250000e+02,1.000000000000000000e+02,9.001937866210937500e+01,9.
000000000000000000e+01,2.104975781250000000e+05
3.265061187744140625e+01,1.008095397949218750e+02,1.000000000000000000e+02,9.001354980468750000e+01,9.
000000000000000000e+01,2.103660781250000000e+05
3.290278792381286621e+01,1.007754516601562500e+02,1.000000000000000000e+02,9.000537109375000000e+01,9.
000000000000000000e+01,2.101731718750000000e+05
3.315201187133789062e+01,1.007527847290039062e+02,1.000000000000000000e+02,9.000028991699218750e+01,9.
000000000000000000e+01,2.100475468750000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
3.340159797668457031e+01,1.007189483642578125e+02,1.000000000000000000e+02,8.999317932128906250e+01,9.
000000000000000000e+01,2.098635937500000000e+05
3.365175485610961914e+01,1.006965408325195312e+02,1.000000000000000000e+02,8.998878479003906250e+01,9.
000000000000000000e+01,2.097440468750000000e+05
3.390092897415161133e+01,1.006632156372070312e+02,1.000000000000000000e+02,8.998265075683593750e+01,9.
000000000000000000e+01,2.095693125000000000e+05
3.415194272994995117e+01,1.006412200927734375e+02,1.000000000000000000e+02,8.997885131835937500e+01,9.
000000000000000000e+01,2.094559843750000000e+05
3.440155887603759766e+01,1.006086349487304688e+02,1.000000000000000000e+02,8.997358703613281250e+01,9.
000000000000000000e+01,2.092906562500000000e+05
3.465157699584960938e+01,1.005872039794921875e+02,1.000000000000000000e+02,8.997033691406250000e+01,9.
000000000000000000e+01,2.091835937500000000e+05
3.490317893028259277e+01,1.005555572509765625e+02,1.000000000000000000e+02,8.996583557128906250e+01,9.
000000000000000000e+01,2.090277656250000000e+05
3.515011191368103027e+01,1.005348129272460938e+02,1.000000000000000000e+02,8.996307373046875000e+01,9.
000000000000000000e+01,2.089270312500000000e+05
3.540221786499023438e+01,1.005042648315429688e+02,1.000000000000000000e+02,8.995925903320312500e+01,9.
000000000000000000e+01,2.087807031250000000e+05
3.565184307098388672e+01,1.004843063354492188e+02,1.000000000000000000e+02,8.995692443847656250e+01,9.
000000000000000000e+01,2.086863125000000000e+05
3.590138602256774902e+01,1.004550094604492188e+02,1.000000000000000000e+02,8.995372009277343750e+01,9.
000000000000000000e+01,2.085494687500000000e+05
3.615191888809204102e+01,1.004359207153320312e+02,1.000000000000000000e+02,8.995176696777343750e+01,9.
000000000000000000e+01,2.084614062500000000e+05
3.640138387680053711e+01,1.004079818725585938e+02,1.000000000000000000e+02,8.994909667968750000e+01,9.
000000000000000000e+01,2.083339531250000000e+05
3.665224909782409668e+01,1.003898315429687500e+02,1.000000000000000000e+02,8.994747924804687500e+01,9.
000000000000000000e+01,2.082520781250000000e+05
3.690253281593322754e+01,1.003633346557617188e+02,1.000000000000000000e+02,8.994527435302734375e+01,9.
000000000000000000e+01,2.081338906250000000e+05
3.715127706527709961e+01,1.003461685180664062e+02,1.000000000000000000e+02,8.994395446777343750e+01,9.
000000000000000000e+01,2.080580781250000000e+05
3.740212774276733398e+01,1.003211898803710938e+02,1.000000000000000000e+02,8.994216156005859375e+01,9.
000000000000000000e+01,2.079489218750000000e+05
3.765139079093933105e+01,1.003050460815429688e+02,1.000000000000000000e+02,8.994108581542968750e+01,9.
000000000000000000e+01,2.078790781250000000e+05
3.790395379066467285e+01,1.002816162109375000e+02,1.000000000000000000e+02,8.993966674804687500e+01,9.
000000000000000000e+01,2.077787031250000000e+05
3.806749272346496582e+01,1.002665176391601562e+02,1.000000000000000000e+02,8.993881225585937500e+01,9.
000000000000000000e+01,2.077464218750000000e+05
3.830170702934265137e+01,1.002518386840820312e+02,1.000000000000000000e+02,8.993804931640625000e+01,9.
000000000000000000e+01,2.076528906250000000e+05
3.855007791519165039e+01,1.002375869750976562e+02,1.000000000000000000e+02,8.993736267089843750e+01,9.
000000000000000000e+01,2.075933281250000000e+05
3.880190086364746094e+01,1.002169952392578125e+02,1.000000000000000000e+02,8.993645477294921875e+01,9.
000000000000000000e+01,2.075080312500000000e+05
3.905206704139709473e+01,1.002037887573242188e+02,1.000000000000000000e+02,8.993592834472656250e+01,9.
000000000000000000e+01,2.074538750000000000e+05
3.930240893363952637e+01,1.001847763061523438e+02,1.000000000000000000e+02,8.993526458740234375e+01,9.
000000000000000000e+01,2.073765312500000000e+05
3.955116701126098633e+01,1.001726150512695312e+02,1.000000000000000000e+02,8.993489074707031250e+01,9.
000000000000000000e+01,2.073275468750000000e+05
3.980261301994323730e+01,1.001551437377929688e+02,1.000000000000000000e+02,8.993441772460937500e+01,9.
000000000000000000e+01,2.072577968750000000e+05
4.005088305473327637e+01,1.001440124511718750e+02,1.000000000000000000e+02,8.993416595458984375e+01,9.
000000000000000000e+01,2.072137343750000000e+05
4.030238270759582520e+01,1.001280593872070312e+02,1.000000000000000000e+02,8.993386840820312500e+01,9.
000000000000000000e+01,2.071511562500000000e+05
4.055114579200744629e+01,1.001179199218750000e+02,1.000000000000000000e+02,8.993372344970703125e+01,9.
000000000000000000e+01,2.071117500000000000e+05
4.080140781402587891e+01,1.001034469604492188e+02,1.000000000000000000e+02,8.993356323242187500e+01,9.
000000000000000000e+01,2.070560156250000000e+05
4.105140089988708496e+01,1.000942764282226562e+02,1.000000000000000000e+02,8.993350219726562500e+01,9.
000000000000000000e+01,2.070209843750000000e+05
4.130235695838928223e+01,1.000812149047851562e+02,1.000000000000000000e+02,8.993347167968750000e+01,9.
000000000000000000e+01,2.069716562500000000e+05
4.155214881896972656e+01,1.000729751586914062e+02,1.000000000000000000e+02,8.993347930908203125e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,2.069407812500000000e+05
4.180314278602600098e+01,1.000612716674804688e+02,1.000000000000000000e+02,8.993354034423828125e+01,9.
000000000000000000e+01,2.068973906250000000e+05
4.205146789550781250e+01,1.000539093017578125e+02,1.000000000000000000e+02,8.993360900878906250e+01,9.
000000000000000000e+01,2.068703906250000000e+05
4.230236697196960449e+01,1.000435104370117188e+02,1.000000000000000000e+02,8.993374633789062500e+01,9.
000000000000000000e+01,2.068326406250000000e+05
4.255170893669128418e+01,1.000369873046875000e+02,1.000000000000000000e+02,8.993386840820312500e+01,9.
000000000000000000e+01,2.068092187500000000e+05
4.280158090591430664e+01,1.000278015136718750e+02,1.000000000000000000e+02,8.993406677246093750e+01,9.
000000000000000000e+01,2.067766250000000000e+05
4.305187773704528809e+01,1.000220718383789062e+02,1.000000000000000000e+02,8.993421936035156250e+01,9.
000000000000000000e+01,2.067565468750000000e+05
4.330140590667724609e+01,1.000140380859375000e+02,1.000000000000000000e+02,8.993446350097656250e+01,9.
000000000000000000e+01,2.067287656250000000e+05
4.355292177200317383e+01,1.000090484619140625e+02,1.000000000000000000e+02,8.993464660644531250e+01,9.
000000000000000000e+01,2.067117031250000000e+05
4.380288887023925781e+01,1.000020904541015625e+02,1.000000000000000000e+02,8.993493652343750000e+01,9.
000000000000000000e+01,2.066883281250000000e+05
4.405108284950256348e+01,9.999779510498046875e+01,1.000000000000000000e+02,8.993513488769531250e+01,9.
000000000000000000e+01,2.066740937500000000e+05
4.430412697792053223e+01,9.999182891845703125e+01,1.000000000000000000e+02,8.993545532226562500e+01,9.
000000000000000000e+01,2.066546718750000000e+05
4.455206489562988281e+01,9.998817443847656250e+01,1.000000000000000000e+02,8.993566131591796875e+01,9.
000000000000000000e+01,2.066430312500000000e+05
4.480179405212402344e+01,9.998313140869140625e+01,1.000000000000000000e+02,8.993599700927734375e+01,9.
000000000000000000e+01,2.066272500000000000e+05
4.505226087570190430e+01,9.998006439208984375e+01,1.000000000000000000e+02,8.993621826171875000e+01,9.
000000000000000000e+01,2.066179062500000000e+05
4.530184698104858398e+01,9.997586822509765625e+01,1.000000000000000000e+02,8.993656158447265625e+01,9.
000000000000000000e+01,2.066054531250000000e+05
4.555228686332702637e+01,9.997333526611328125e+01,1.000000000000000000e+02,8.993679809570312500e+01,9.
000000000000000000e+01,2.065981875000000000e+05
4.580272483825683594e+01,9.996991729736328125e+01,1.000000000000000000e+02,8.993713378906250000e+01,9.
000000000000000000e+01,2.065887031250000000e+05
4.605080890655517578e+01,9.996787261962890625e+01,1.000000000000000000e+02,8.993737030029296875e+01,9.
000000000000000000e+01,2.065832656250000000e+05
4.630244588851928711e+01,9.996514892578125000e+01,1.000000000000000000e+02,8.993771362304687500e+01,9.
000000000000000000e+01,2.065764218750000000e+05
4.655220079421997070e+01,9.996354675292968750e+01,1.000000000000000000e+02,8.993794250488281250e+01,9.
000000000000000000e+01,2.065726562500000000e+05
4.680287575721740723e+01,9.996145629882812500e+01,1.000000000000000000e+02,8.993828582763671875e+01,9.
000000000000000000e+01,2.065681250000000000e+05
4.705172181129455566e+01,9.996025085449218750e+01,1.000000000000000000e+02,8.993850708007812500e+01,9.
000000000000000000e+01,2.065658437500000000e+05
4.730165505409240723e+01,9.995871734619140625e+01,1.000000000000000000e+02,8.993884277343750000e+01,9.
000000000000000000e+01,2.065633906250000000e+05
4.755240797996520996e+01,9.995787048339843750e+01,1.000000000000000000e+02,8.993905639648437500e+01,9.
000000000000000000e+01,2.065623437500000000e+05
4.780261301994323730e+01,9.995684051513671875e+01,1.000000000000000000e+02,8.993937683105468750e+01,9.
000000000000000000e+01,2.065616562500000000e+05
4.801543688774108887e+01,9.995630645751953125e+01,1.000000000000000000e+02,8.993959045410156250e+01,9.
000000000000000000e+01,2.065617187500000000e+05
4.825107192993164062e+01,9.995588684082031250e+01,1.000000000000000000e+02,8.993980407714843750e+01,9.
000000000000000000e+01,2.065621875000000000e+05
4.850313997268676758e+01,9.995545196533203125e+01,1.000000000000000000e+02,8.994010162353515625e+01,9.
000000000000000000e+01,2.065635937500000000e+05
4.875200891494750977e+01,9.995529174804687500e+01,1.000000000000000000e+02,8.994029998779296875e+01,9.
000000000000000000e+01,2.065649687500000000e+05
4.900275611877441406e+01,9.995521545410156250e+01,1.000000000000000000e+02,8.994058227539062500e+01,9.
000000000000000000e+01,2.065675937500000000e+05
4.925014972686767578e+01,9.995527648925781250e+01,1.000000000000000000e+02,8.994077301025390625e+01,9.
000000000000000000e+01,2.065696718750000000e+05
4.950200080871582031e+01,9.995552062988281250e+01,1.000000000000000000e+02,8.994104003906250000e+01,9.
000000000000000000e+01,2.065733437500000000e+05
4.975204992294311523e+01,9.995577239990234375e+01,1.000000000000000000e+02,8.994122314453125000e+01,9.
000000000000000000e+01,2.065760312500000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
5.000288391113281250e+01,9.995627593994140625e+01,1.000000000000000000e+02,8.994148254394531250e+01,9.
000000000000000000e+01,2.065805312500000000e+05
5.025228095054626465e+01,9.995668792724609375e+01,1.000000000000000000e+02,8.994164276123046875e+01,9.
000000000000000000e+01,2.065837500000000000e+05
5.050313901901245117e+01,9.995740509033203125e+01,1.000000000000000000e+02,8.994188690185546875e+01,9.
000000000000000000e+01,2.065888593750000000e+05
5.075468182563781738e+01,9.995795440673828125e+01,1.000000000000000000e+02,8.994203948974609375e+01,9.
000000000000000000e+01,2.065924843750000000e+05
5.100411486625671387e+01,9.995885467529296875e+01,1.000000000000000000e+02,8.994226074218750000e+01,9.
000000000000000000e+01,2.065981406250000000e+05
5.125216674804687500e+01,9.995951080322265625e+01,1.000000000000000000e+02,8.994241333007812500e+01,9.
000000000000000000e+01,2.066020468750000000e+05
5.150267100334167480e+01,9.996055603027343750e+01,1.000000000000000000e+02,8.994261169433593750e+01,9.
000000000000000000e+01,2.066080781250000000e+05
5.175315594673156738e+01,9.996129608154296875e+01,1.000000000000000000e+02,8.994274902343750000e+01,9.
000000000000000000e+01,2.066121875000000000e+05
5.200182294845581055e+01,9.996244812011718750e+01,1.000000000000000000e+02,8.994294738769531250e+01,9.
000000000000000000e+01,2.066185000000000000e+05
5.226125001907348633e+01,9.996324920654296875e+01,1.000000000000000000e+02,8.994306945800781250e+01,9.
000000000000000000e+01,2.066227500000000000e+05
5.250283002853393555e+01,9.996449279785156250e+01,1.000000000000000000e+02,8.994325256347656250e+01,9.
000000000000000000e+01,2.066291875000000000e+05
5.275174689292907715e+01,9.996533966064453125e+01,1.000000000000000000e+02,8.994335937500000000e+01,9.
000000000000000000e+01,2.066334843750000000e+05
5.300342988967895508e+01,9.996663665771484375e+01,1.000000000000000000e+02,8.994351959228515625e+01,9.
000000000000000000e+01,2.066400000000000000e+05
5.325051593780517578e+01,9.996751403808593750e+01,1.000000000000000000e+02,8.994361877441406250e+01,9.
000000000000000000e+01,2.066443437500000000e+05
5.350221276283264160e+01,9.996884155273437500e+01,1.000000000000000000e+02,8.994377136230468750e+01,9.
000000000000000000e+01,2.066508281250000000e+05
5.375315403938293457e+01,9.996973419189453125e+01,1.000000000000000000e+02,8.994386291503906250e+01,9.
000000000000000000e+01,2.066551093750000000e+05
5.400236797332763672e+01,9.997107696533203125e+01,1.000000000000000000e+02,8.994400024414062500e+01,9.
000000000000000000e+01,2.066615312500000000e+05
5.425300979614257812e+01,9.997197723388671875e+01,1.000000000000000000e+02,8.994409179687500000e+01,9.
000000000000000000e+01,2.066657656250000000e+05
5.450370979309082031e+01,9.997331237792968750e+01,1.000000000000000000e+02,8.994421386718750000e+01,9.
000000000000000000e+01,2.066720000000000000e+05
5.475102901458740234e+01,9.997419738769531250e+01,1.000000000000000000e+02,8.994429016113281250e+01,9.
000000000000000000e+01,2.066760937500000000e+05
5.500390291213989258e+01,9.997552490234375000e+01,1.000000000000000000e+02,8.994440460205078125e+01,9.
000000000000000000e+01,2.066821875000000000e+05
5.525131487846374512e+01,9.997639465332031250e+01,1.000000000000000000e+02,8.994447326660156250e+01,9.
000000000000000000e+01,2.066861406250000000e+05
5.550271368026733398e+01,9.997768402099609375e+01,1.000000000000000000e+02,8.994458007812500000e+01,9.
000000000000000000e+01,2.066919531250000000e+05
5.575385594367980957e+01,9.997853088378906250e+01,1.000000000000000000e+02,8.994464111328125000e+01,9.
000000000000000000e+01,2.066957500000000000e+05
5.600253677368164062e+01,9.997978210449218750e+01,1.000000000000000000e+02,8.994473266601562500e+01,9.
000000000000000000e+01,2.067013281250000000e+05
5.625252079963684082e+01,9.998059844970703125e+01,1.000000000000000000e+02,8.994479370117187500e+01,9.
000000000000000000e+01,2.067049531250000000e+05
5.650333595275878906e+01,9.998179626464843750e+01,1.000000000000000000e+02,8.994486999511718750e+01,9.
000000000000000000e+01,2.067102500000000000e+05
5.675358104705810547e+01,9.998258209228515625e+01,1.000000000000000000e+02,8.994491577148437500e+01,9.
000000000000000000e+01,2.067136250000000000e+05
5.700284576416015625e+01,9.998372650146484375e+01,1.000000000000000000e+02,8.994499206542968750e+01,9.
000000000000000000e+01,2.067186250000000000e+05
5.725124096870422363e+01,9.998446655273437500e+01,1.000000000000000000e+02,8.994503784179687500e+01,9.
000000000000000000e+01,2.067218437500000000e+05
5.750281500816345215e+01,9.998554992675781250e+01,1.000000000000000000e+02,8.994509887695312500e+01,9.
000000000000000000e+01,2.067265156250000000e+05
5.775190091133117676e+01,9.998625183105468750e+01,1.000000000000000000e+02,8.994514465332031250e+01,9.
000000000000000000e+01,2.067295156250000000e+05
5.800270986557006836e+01,9.998727416992187500e+01,1.000000000000000000e+02,8.994519805908203125e+01,9.
000000000000000000e+01,2.067338437500000000e+05
5.820328497886657715e+01,9.998793029785156250e+01,1.000000000000000000e+02,8.994523620605468750e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,2.067366562500000000e+05
5.845084476470947266e+01,9.998857116699218750e+01,1.000000000000000000e+02,8.994526672363281250e+01,9.
000000000000000000e+01,2.067393281250000000e+05
5.870220184326171875e+01,9.998950195312500000e+01,1.000000000000000000e+02,8.994532012939453125e+01,9.
000000000000000000e+01,2.067432500000000000e+05
5.895184397697448730e+01,9.999009704589843750e+01,1.000000000000000000e+02,8.994534301757812500e+01,9.
000000000000000000e+01,2.067457500000000000e+05
5.920318269729614258e+01,9.999095153808593750e+01,1.000000000000000000e+02,8.994538879394531250e+01,9.
000000000000000000e+01,2.067493125000000000e+05
5.945280981063842773e+01,9.999150085449218750e+01,1.000000000000000000e+02,8.994541931152343750e+01,9.
000000000000000000e+01,2.067515781250000000e+05
5.970341777801513672e+01,9.999229431152343750e+01,1.000000000000000000e+02,8.994544982910156250e+01,9.
000000000000000000e+01,2.067548750000000000e+05
5.995184898376464844e+01,9.999279785156250000e+01,1.000000000000000000e+02,8.994548034667968750e+01,9.
000000000000000000e+01,2.067569531250000000e+05
6.020380997657775879e+01,9.999352264404296875e+01,1.000000000000000000e+02,8.994551086425781250e+01,9.
000000000000000000e+01,2.067599062500000000e+05
6.045096087455749512e+01,9.999398803710937500e+01,1.000000000000000000e+02,8.994552612304687500e+01,9.
000000000000000000e+01,2.067617656250000000e+05
6.070328903198242188e+01,9.999464416503906250e+01,1.000000000000000000e+02,8.994555664062500000e+01,9.
000000000000000000e+01,2.067644531250000000e+05
6.095240187644958496e+01,9.999506378173828125e+01,1.000000000000000000e+02,8.994557189941406250e+01,9.
000000000000000000e+01,2.067661562500000000e+05
6.120274877548217773e+01,9.999565887451171875e+01,1.000000000000000000e+02,8.994560241699218750e+01,9.
000000000000000000e+01,2.067685468750000000e+05
6.145317077636718750e+01,9.999603271484375000e+01,1.000000000000000000e+02,8.994561767578125000e+01,9.
000000000000000000e+01,2.067700781250000000e+05
6.170326709747314453e+01,9.999656677246093750e+01,1.000000000000000000e+02,8.994563293457031250e+01,9.
000000000000000000e+01,2.067722031250000000e+05
6.195357584953308105e+01,9.999691009521484375e+01,1.000000000000000000e+02,8.994564819335937500e+01,9.
000000000000000000e+01,2.067735468750000000e+05
6.220450568199157715e+01,9.999738311767578125e+01,1.000000000000000000e+02,8.994566345214843750e+01,9.
000000000000000000e+01,2.067754375000000000e+05
6.245298099517822266e+01,9.999768829345703125e+01,1.000000000000000000e+02,8.994567108154296875e+01,9.
000000000000000000e+01,2.067766250000000000e+05
6.270364093780517578e+01,9.999810791015625000e+01,1.000000000000000000e+02,8.994569396972656250e+01,9.
000000000000000000e+01,2.067782968750000000e+05
6.295130181312561035e+01,9.999836730957031250e+01,1.000000000000000000e+02,8.994569396972656250e+01,9.
000000000000000000e+01,2.067793125000000000e+05
6.320295381546020508e+01,9.999874114990234375e+01,1.000000000000000000e+02,8.994570922851562500e+01,9.
000000000000000000e+01,2.067807343750000000e+05
6.345245695114135742e+01,9.999897003173828125e+01,1.000000000000000000e+02,8.994571685791015625e+01,9.
000000000000000000e+01,2.067816250000000000e+05
6.370289778709411621e+01,9.999929809570312500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067828750000000000e+05
6.395397281646728516e+01,9.999949645996093750e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067836406250000000e+05
6.420247697830200195e+01,9.999977111816406250e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067847187500000000e+05
6.445560789108276367e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067853281250000000e+05
6.470355606079101562e+01,1.000001754760742188e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067862500000000000e+05
6.495266771316528320e+01,1.000003204345703125e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067867500000000000e+05
6.520334386825561523e+01,1.000005187988281250e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067874843750000000e+05
6.545144486427307129e+01,1.000006408691406250e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067879531250000000e+05
6.570318889617919922e+01,1.000008010864257812e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067885156250000000e+05
6.595331978797912598e+01,1.000009002685546875e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067888906250000000e+05
6.620307683944702148e+01,1.000010375976562500e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067893437500000000e+05
6.645381188392639160e+01,1.000011138916015625e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067896093750000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
6.670452284812927246e+01,1.000012207031250000e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067899843750000000e+05
6.695256400108337402e+01,1.000012817382812500e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067901718750000000e+05
6.720303297042846680e+01,1.000013580322265625e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067904218750000000e+05
6.745380973815917969e+01,1.000014038085937500e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067905625000000000e+05
6.770306301116943359e+01,1.000014572143554688e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067907343750000000e+05
6.795338010787963867e+01,1.000014953613281250e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067908281250000000e+05
6.810334277153015137e+01,1.000015335083007812e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067908906250000000e+05
6.835350871086120605e+01,1.000015411376953125e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067909375000000000e+05
6.860374975204467773e+01,1.000015640258789062e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067909843750000000e+05
6.885370588302612305e+01,1.000015716552734375e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067909843750000000e+05
6.910527491569519043e+01,1.000015792846679688e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067909843750000000e+05
6.935150003433227539e+01,1.000015792846679688e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067909531250000000e+05
6.960391974449157715e+01,1.000015792846679688e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067908906250000000e+05
6.985235285758972168e+01,1.000015716552734375e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067908437500000000e+05
7.010346078872680664e+01,1.000015563964843750e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067907500000000000e+05
7.035332608222961426e+01,1.000015411376953125e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067906718750000000e+05
7.060396099090576172e+01,1.000015182495117188e+02,1.000000000000000000e+02,8.994577789306640625e+01,9.
000000000000000000e+01,2.067905000000000000e+05
7.085444092750549316e+01,1.000014953613281250e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067904062500000000e+05
7.110599708557128906e+01,1.000014648437500000e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067902500000000000e+05
7.135161590576171875e+01,1.000014419555664062e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067901406250000000e+05
7.160401082038879395e+01,1.000014038085937500e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067899687500000000e+05
7.185275578498840332e+01,1.000013732910156250e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067898437500000000e+05
7.210337185859680176e+01,1.000013351440429688e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067896250000000000e+05
7.235331082344055176e+01,1.000013046264648438e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067895000000000000e+05
7.260422706604003906e+01,1.000012588500976562e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067892968750000000e+05
7.285370397567749023e+01,1.000012283325195312e+02,1.000000000000000000e+02,8.994577026367187500e+01,9.
000000000000000000e+01,2.067891562500000000e+05
7.310391283035278320e+01,1.000011749267578125e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067889218750000000e+05
7.335335969924926758e+01,1.000011444091796875e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067887968750000000e+05
7.360401582717895508e+01,1.000010986328125000e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067885781250000000e+05
7.385320806503295898e+01,1.000010604858398438e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067884375000000000e+05
7.410352396965026855e+01,1.000010147094726562e+02,1.000000000000000000e+02,8.994576263427734375e+01,9.
000000000000000000e+01,2.067882343750000000e+05
7.435293698310852051e+01,1.000009765625000000e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067880625000000000e+05
7.460405087471008301e+01,1.000009307861328125e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067878593750000000e+05
7.485331606864929199e+01,1.000009002685546875e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
```

```
000000000000000000e+01,2.067877343750000000e+05
7.510460495948791504e+01,1.000008468627929688e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067875312500000000e+05
7.535340380668640137e+01,1.000008163452148438e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067873906250000000e+05
7.560458588600158691e+01,1.000007705688476562e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067871718750000000e+05
7.585157299041748047e+01,1.000007400512695312e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067870468750000000e+05
7.610368490219116211e+01,1.000006942749023438e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067868593750000000e+05
7.635263085365295410e+01,1.000006637573242188e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067867500000000000e+05
7.660352087020874023e+01,1.000006179809570312e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067865625000000000e+05
7.685431885719299316e+01,1.000005874633789062e+02,1.000000000000000000e+02,8.994575500488281250e+01,9.
000000000000000000e+01,2.067864531250000000e+05
7.710354089736938477e+01,1.000005493164062500e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067862656250000000e+05
7.735434699058532715e+01,1.000005187988281250e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067861562500000000e+05
7.760430002212524414e+01,1.000004806518554688e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067860000000000000e+05
7.785213279724121094e+01,1.000004577636718750e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067858906250000000e+05
7.806637382507324219e+01,1.000004196166992188e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067857968750000000e+05
7.830354690551757812e+01,1.000003967285156250e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067856562500000000e+05
7.855472087860107422e+01,1.000003738403320312e+02,1.000000000000000000e+02,8.994574737548828125e+01,9.
000000000000000000e+01,2.067855625000000000e+05
7.880455398559570312e+01,1.000003356933593750e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067854062500000000e+05
7.905227494239807129e+01,1.000003204345703125e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067853281250000000e+05
7.930394601821899414e+01,1.000002899169921875e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067852031250000000e+05
7.955242991447448730e+01,1.000002670288085938e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067851406250000000e+05
7.980347895622253418e+01,1.000002441406250000e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067850312500000000e+05
8.005338191986083984e+01,1.000002212524414062e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067849531250000000e+05
8.030325078964233398e+01,1.000001983642578125e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067848593750000000e+05
8.055377197265625000e+01,1.000001831054687500e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067847968750000000e+05
8.080461192131042480e+01,1.000001602172851562e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067847187500000000e+05
8.105306482315063477e+01,1.000001449584960938e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067846562500000000e+05
8.130466389656066895e+01,1.000001220703125000e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067845468750000000e+05
8.155292797088623047e+01,1.000001068115234375e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067845000000000000e+05
8.180403494834899902e+01,1.000000915527343750e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067844375000000000e+05
8.205550694465637207e+01,1.000000839233398438e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067843906250000000e+05
8.230448985099792480e+01,1.000000610351562500e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067843281250000000e+05
8.255410099029541016e+01,1.000000534057617188e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067842968750000000e+05
8.280450510978698730e+01,1.000000381469726562e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067842343750000000e+05
8.305241894721984863e+01,1.000000305175781250e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067842031250000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
8.330489492416381836e+01,1.000000228881835938e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067841562500000000e+05
8.355332994461059570e+01,1.000000152587890625e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067841406250000000e+05
8.380411100387573242e+01,1.000000000000000000e+02,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067840937500000000e+05
8.405337071418762207e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067840781250000000e+05
8.430362892150878906e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067840468750000000e+05
8.455403494834899902e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067840312500000000e+05
8.480468392372131348e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067840000000000000e+05
8.505363082885742188e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839843750000000e+05
8.530398488044738770e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839531250000000e+05
8.555430603027343750e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839531250000000e+05
8.580398678779602051e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839375000000000e+05
8.605551695823669434e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839218750000000e+05
8.630180668830871582e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839062500000000e+05
8.650554800033569336e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067839062500000000e+05
8.675258278846740723e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994573974609375000e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.700403189659118652e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.725421786308288574e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.750388383865356445e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.775374388694763184e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.800375485420227051e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.815422582626342773e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.840374803543090820e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.865399694442749023e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.890461683273315430e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.915255689620971680e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.940475988388061523e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838593750000000e+05
8.965275287628173828e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838750000000000e+05
8.990366697311401367e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838750000000000e+05
9.015490674972534180e+01,9.999993896484375000e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838750000000000e+05
9.040407872200012207e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838906250000000e+05
9.065494489669799805e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838906250000000e+05
9.090463304519653320e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067838906250000000e+05
9.115366697311401367e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839062500000000e+05
9.140520787239074707e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,2.067839062500000000e+05
9.165261697769165039e+01,9.999994659423828125e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839062500000000e+05
9.190412902832031250e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839218750000000e+05
9.215438199043273926e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839218750000000e+05
9.240428495407104492e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839375000000000e+05
9.265486693382263184e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839375000000000e+05
9.290510296821594238e+01,9.999995422363281250e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839531250000000e+05
9.315437579154968262e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839531250000000e+05
9.340530204772949219e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839687500000000e+05
9.365431094169616699e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839687500000000e+05
9.390453767776489258e+01,9.999996185302734375e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839687500000000e+05
9.415500187873840332e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839843750000000e+05
9.440447592735290527e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067839843750000000e+05
9.465575385093688965e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840000000000000e+05
9.490533590316772461e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840000000000000e+05
9.515421581268310547e+01,9.999996948242187500e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840000000000000e+05
9.540473079681396484e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840156250000000e+05
9.565366506576538086e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840156250000000e+05
9.590437698364257812e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840312500000000e+05
9.615503382682800293e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840312500000000e+05
9.640442991256713867e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840312500000000e+05
9.665530991554260254e+01,9.999997711181640625e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840468750000000e+05
9.690575790405273438e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840468750000000e+05
9.715504980087280273e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840468750000000e+05
9.740516400337219238e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840625000000000e+05
9.765476608276367188e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840625000000000e+05
9.790490078926086426e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840625000000000e+05
9.810453772544860840e+01,9.999998474121093750e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840625000000000e+05
9.835479974746704102e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840781250000000e+05
9.860491800308227539e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840781250000000e+05
9.885598778724670410e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840781250000000e+05
9.910740184783935547e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840781250000000e+05
9.935448908805847168e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840781250000000e+05
9.960474586486816406e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
9.985504412651062012e+01,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
1.001043539047241211e+02,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
1.003553137779235840e+02,9.999999237060546875e+01,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
1.006048638820648193e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
1.008552188873291016e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067840937500000000e+05
1.011053678989410400e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.013547346591949463e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.016057851314544678e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.018528540134429932e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.021042187213897705e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.023553678989410400e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.026052558422088623e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.028557927608489990e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.031049368381500244e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.033547677993774414e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.036049978733062744e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.038552279472351074e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.041059279441833496e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.043549177646636963e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.046048450469970703e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.048554539680480957e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.051043620109558105e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.053559007644653320e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.056062047481536865e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.058540170192718506e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.061056430339813232e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.063536367416381836e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.066047317981719971e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.068560068607330322e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.071046907901763916e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.073552079200744629e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.076054708957672119e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.078555619716644287e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.080209977626800537e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
000000000000000000e+01,2.067841250000000000e+05
1.082552139759063721e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.085048739910125732e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.087551600933074951e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.090062668323516846e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.092553730010986328e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.095049767494201660e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.097550539970397949e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.100049147605895996e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.102553279399871826e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.105058259963989258e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.107545657157897949e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.110056688785552979e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.112538397312164307e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.115051689147949219e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.117544808387756348e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.120059108734130859e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.122563948631286621e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.125060060024261475e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.127560529708862305e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.130070137977600098e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841250000000000e+05
1.132542488574981689e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.135064229965209961e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.137542719841003418e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.140053029060363770e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.142563397884368896e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.145055408477783203e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.147553720474243164e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.150065929889678955e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.152536969184875488e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.155063080787658691e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.157539608478546143e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.160054128170013428e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.162565050125122070e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_step_90pitch_and_100rpm.csv

```
1.165078148841857910e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.167560687065124512e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.170067248344421387e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.172537488937377930e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.175065116882324219e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.177634899616241455e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.180080418586730957e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.182062649726867676e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.184555809497833252e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.187065289020538330e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.189552156925201416e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.192056939601898193e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.194561738967895508e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.197052810192108154e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.199557707309722900e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.202061817646026611e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.204555919170379639e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.207070209980010986e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.209551308155059814e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.212062468528747559e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.214538748264312744e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.217052338123321533e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.219565989971160889e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.222059519290924072e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.224570739269256592e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.227063689231872559e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.229131550788879395e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.231066758632659912e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.233542699813842773e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.236063969135284424e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
1.238562560081481934e+02,1.000000000000000000e+02,1.000000000000000000e+02,8.994572448730468750e+01,9.
000000000000000000e+01,2.067841093750000000e+05
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
2.005369663238525391e-01,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
4.474139213562011719e-01,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
6.096739768981933594e-01,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
8.474249839782714844e-01,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
1.049288988113403320e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
1.248259067535400391e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
1.498853206634521484e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
1.749516010284423828e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
1.951392889022827148e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
2.150790929794311523e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
2.351185083389282227e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
2.598169088363647461e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
2.849444150924682617e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
3.048119068145751953e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
3.247919082641601562e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
3.448205947875976562e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
3.702715873718261719e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
3.948809862136840820e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
4.200562953948974609e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
4.448179006576538086e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
4.648663043975830078e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
4.899060964584350586e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
5.148450136184692383e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
5.348732948303222656e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
5.597250938415527344e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
5.848742008209228516e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
6.098527908325195312e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
6.348805189132690430e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
6.549146175384521484e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.000000000000000000e+00,0.
000000000000000000e+00,0.000000000000000000e+00
6.798204898834228516e+00,1.000761566162109375e+02,0.000000000000000000e+00,6.802572752349078655e-04,3.
401168063282966614e-02,2.720999531447887421e-03
6.999503135681152344e+00,1.000507736206054688e+02,0.000000000000000000e+00,6.122167687863111496e-03,6.
121841818094253540e-02,1.088370196521282196e-02
7.200576066970825195e+00,1.000761566162109375e+02,0.000000000000000000e+00,2.448763325810432434e-02,8.
842279016971588135e-02,2.448763325810432434e-02
7.448368787765502930e+00,1.000761566162109375e+02,0.000000000000000000e+00,4.353232309222221375e-02,1.
156247854232788086e-01,5.509480088949203491e-02
7.648407936096191406e+00,1.000761566162109375e+02,0.000000000000000000e+00,6.801728904247283936e-02,1.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
428244262933731079e-01,8.229973167181015015e-02
7.849371194839477539e+00,1.000761566162109375e+02,0.000000000000000000e+00,9.794206917285919189e-02,1.
700216829776763916e-01,1.149442344903945923e-01
8.045845031738281250e+00,1.000761566162109375e+02,0.000000000000000000e+00,1.333061754703521729e-01,1.
972165852785110474e-01,1.530278325080871582e-01
8.249591112136840820e+00,1.000761566162109375e+02,0.000000000000000000e+00,1.741091459989547729e-01,2.
244091033935546875e-01,1.965500563383102417e-01
8.498257875442504883e+00,1.000761566162109375e+02,0.000000000000000000e+00,2.203150689601898193e-01,2.
619401812553405762e-01,2.453311979770660400e-01
8.748912096023559570e+00,1.000761566162109375e+02,0.000000000000000000e+00,2.988274395465850830e-01,2.
834427952766418457e-01,3.271717131137847900e-01
8.950685977935791016e+00,1.000761566162109375e+02,0.000000000000000000e+00,3.564952909946441650e-01,3.
110617399215698242e-01,3.867383599281311035e-01
9.151418209075927734e+00,1.000761566162109375e+02,0.000000000000000000e+00,4.178445339202880859e-01,3.
267568349838256836e-01,4.497607350349426270e-01
9.397714853286743164e+00,1.000761566162109375e+02,0.000000000000000000e+00,4.824364185333251953e-01,3.
405381739139556885e-01,5.158237814903259277e-01
9.601245880126953125e+00,1.000761566162109375e+02,0.000000000000000000e+00,5.845551490783691406e-01,3.
526096045970916748e-01,5.845551490783691406e-01
9.846120834350585938e+00,1.000761566162109375e+02,0.000000000000000000e+00,6.556223630905151367e-01,3.
631568551063537598e-01,6.919381022453308105e-01
1.000058698654174805e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.287293076515197754e-01,3.
723484277725219727e-01,7.287293076515197754e-01
1.024835300445556641e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.036125302314758301e-01,3.
803369104862213135e-01,8.416462540626525879e-01
1.049881005287170410e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.800386190414428711e-01,3.
903615772724151611e-01,9.187645912170410156e-01
1.065212893486022949e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.578007459640502930e-01,3.
932422399520874023e-01,9.971249699592590332e-01
1.089794516563415527e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.036716222763061523e+00,4.
006858468055725098e-01,1.076555132865905762e+00
1.114806199073791504e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.156904935836791992e+00,4.
047788977622985840e-01,1.197382807731628418e+00
1.134864401817321777e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.238042473793029785e+00,4.
082730114459991455e-01,1.278869748115539551e+00
1.159811305999755859e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.319851636886596680e+00,4.
125539064407348633e-01,1.360975980758666992e+00
1.184869289398193359e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.443607211112976074e+00,4.
148627221584320068e-01,1.485093474388122559e+00
1.209948396682739258e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.526681065559387207e+00,4.
176486134529113770e-01,1.568361163139343262e+00
1.234830784797668457e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.651968121528625488e+00,4.
191245138645172119e-01,1.693880558013916016e+00
1.254880905151367188e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.735857009887695312e+00,4.
203441143035888672e-01,1.777891397476196289e+00
1.279775404930114746e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.819978356361389160e+00,4.
217708110809326172e-01,1.862112760543823242e+00
1.300118803977966309e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.946505308151245117e+00,4.
224983155727386475e-01,1.946505308151245117e+00
1.324994993209838867e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.031035661697387695e+00,4.
230746626853942871e-01,2.073343038558959961e+00
1.349763512611389160e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.115674257278442383e+00,4.
237123727798461914e-01,2.158026933670043945e+00
1.374854087829589844e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.242785692214965820e+00,4.
240148961544036865e-01,2.285187244415283203e+00
1.399797201156616211e+01,1.000507736206054688e+02,0.000000000000000000e+00,2.327600240707397461e+00,4.
243092238903045654e-01,2.370023012161254883e+00
1.419949197769165039e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.412453889846801758e+00,4.
244353771209716797e-01,2.454891920089721680e+00
1.435108995437622070e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.497335433959960938e+00,4.
245082736015319824e-01,2.539783239364624023e+00
1.455042409896850586e+01,1.000507736206054688e+02,0.000000000000000000e+00,2.582234144210815430e+00,4.
245273470878601074e-01,2.624686956405639648e+00
1.479833793640136719e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.667140960693359375e+00,4.
245431423187255859e-01,2.709595441818237305e+00
1.504963612556457520e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.794503211975097656e+00,4.
245195984840393066e-01,2.836955070495605469e+00
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
1.524935007095336914e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.879405260086059570e+00,4.
244756400585174561e-01,2.921852827072143555e+00
1.544881796836853027e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.964297533035278320e+00,4.
244160950183868408e-01,3.006739139556884766e+00
1.569947218894958496e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.049177169799804688e+00,4.
243001639842987061e-01,3.091611385345458984e+00
1.594903802871704102e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.176467418670654297e+00,4.
242184162139892578e-01,3.218889236450195312e+00
1.619993019104003906e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.261306524276733398e+00,4.
240827858448028564e-01,3.303719282150268555e+00
1.640117096900939941e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.388531446456909180e+00,4.
239948391914367676e-01,3.388531446456909180e+00
1.655118513107299805e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.430930852890014648e+00,4.
239039421081542969e-01,3.473325967788696289e+00
1.674830889701843262e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.515716314315795898e+00,4.
238574504852294922e-01,3.558102130889892578e+00
1.694863891601562500e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.600483179092407227e+00,4.
237663745880126953e-01,3.642859935760498047e+00
1.714826202392578125e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.685232162475585938e+00,4.
236753880977630615e-01,3.727599620819091797e+00
1.740233898162841797e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.769963026046752930e+00,4.
235568642616271973e-01,3.812322378158569336e+00
1.765038204193115234e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.897029876708984375e+00,4.
234785437583923340e-01,3.939377546310424805e+00
1.784884715080261230e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.981722116470336914e+00,4.
234105646610260010e-01,4.024063110351562500e+00
1.804890799522399902e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.066400527954101562e+00,4.
233404994010925293e-01,4.108734607696533203e+00
1.829749083518981934e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.151065826416015625e+00,4.
232465922832489014e-01,4.193393230438232422e+00
1.850140690803527832e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.278039932250976562e+00,4.
231873452663421631e-01,4.278039932250976562e+00
1.874907302856445312e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.362674713134765625e+00,4.
231384098529815674e-01,4.404988765716552734e+00
1.899836897850036621e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.447299957275390625e+00,4.
230717718601226807e-01,4.489609241485595703e+00
1.924805188179016113e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.574221611022949219e+00,4.
230315089225769043e-01,4.616524696350097656e+00
1.945004105567932129e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.658826351165771484e+00,4.
229913353919982910e-01,4.701125144958496094e+00
1.965206694602966309e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.743422508239746094e+00,4.
229355752468109131e-01,4.785717964172363281e+00
1.989895081520080566e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.828011512756347656e+00,4.
229032993316650391e-01,4.870303153991699219e+00
2.014873814582824707e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.954882621765136719e+00,4.
228768050670623779e-01,4.997169971466064453e+00
2.034924197196960449e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.039456367492675781e+00,4.
228525459766387939e-01,5.081741809844970703e+00
2.056103897094726562e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.124025821685791016e+00,4.
228199720382690430e-01,5.166308879852294922e+00
2.074921393394470215e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.208590984344482422e+00,4.
228070080280303955e-01,5.250871658325195312e+00
2.099965310096740723e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.293150901794433594e+00,4.
227712452411651611e-01,5.335429191589355469e+00
2.124859905242919922e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.419982910156250000e+00,4.
227560758590698242e-01,5.462258338928222656e+00
2.145160794258117676e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.504533290863037109e+00,4.
227328002452850342e-01,5.546807765960693359e+00
2.169889903068542480e+01,1.000507736206054688e+02,0.000000000000000000e+00,5.589080810546875000e+00,4.
227179586887359619e-01,5.631353378295898438e+00
2.194876217842102051e+01,1.000507736206054688e+02,0.000000000000000000e+00,5.715896129608154297e+00,4.
227014183998107910e-01,5.758166313171386719e+00
2.215195703506469727e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.800435543060302734e+00,4.
226849973201751709e-01,5.842704772949218750e+00
2.234843993186950684e+01,1.000507736206054688e+02,0.000000000000000000e+00,5.884973526000976562e+00,4.
226779043674468994e-01,5.927241325378417969e+00
2.255184006690979004e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.969508647918701172e+00,4.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
226613640785217285e-01,6.011775493621826172e+00
2.274874687194824219e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.054041385650634766e+00,4.
226583242416381836e-01,6.096307277679443359e+00
2.299847388267517090e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.138573169708251953e+00,4.
226501584053039551e-01,6.180838108062744141e+00
2.324859380722045898e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.265367984771728516e+00,4.
226451218128204346e-01,6.307632446289062500e+00
2.345001101493835449e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.671153008937835693e-02,4.
226373136043548584e-01,6.392160415649414062e+00
2.369837307929992676e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.512384712696075439e-01,4.
226281344890594482e-01,6.476686954498291016e+00
2.394909787178039551e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.780269086360931396e-01,4.
226244688034057617e-01,6.603474617004394531e+00
2.414926695823669434e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.625516295433044434e-01,4.
226208925247192383e-01,6.687999248504638672e+00
2.439986705780029297e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.470756351947784424e-01,4.
226157367229461670e-01,6.772522449493408203e+00
2.464870595932006836e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.738603472709655762e-01,4.
226123988628387451e-01,6.899306774139404297e+00
2.485129809379577637e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.583826541900634766e-01,4.
226090908050537109e-01,6.983829021453857422e+00
2.509912180900573730e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.429043054580688477e-01,4.
226041734218597412e-01,7.068350315093994141e+00
2.529929184913635254e+01,1.000507736206054688e+02,0.000000000000000000e+00,8.274253010749816895e-01,4.
225949048995971680e-01,7.152870655059814453e+00
2.554944396018981934e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.542041420936584473e-01,4.
225922226905822754e-01,7.279648780822753906e+00
2.574863409996032715e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.038722157478332520e+00,4.
225835204124450684e-01,7.364165782928466797e+00
2.599824786186218262e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.123238444328308105e+00,4.
225744605064392090e-01,7.448681354522705078e+00
2.624914717674255371e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.250010490417480469e+00,4.
225668013095855713e-01,7.575452327728271484e+00
2.644921708106994629e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.334523677825927734e+00,4.
225624799728393555e-01,7.659965515136718750e+00
2.664813995361328125e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.419035911560058594e+00,4.
225583672523498535e-01,7.744476795196533203e+00
2.685001492500305176e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.503547549247741699e+00,4.
225574433803558350e-01,7.828988552093505859e+00
2.709873604774475098e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.588058948516845703e+00,4.
225556850433349609e-01,7.913499832153320312e+00
2.729942607879638672e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.672570109367370605e+00,4.
225542545318603516e-01,7.998011112213134766e+00
2.754867506027221680e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.799336433410644531e+00,4.
225526750087738037e-01,8.124776840209960938e+00
2.775008511543273926e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.883846879005432129e+00,4.
225509166717529297e-01,8.209287643432617188e+00
2.800013589859008789e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.968356966972351074e+00,4.
225419461727142334e-01,8.293796539306640625e+00
2.820203208923339844e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.095119953155517578e+00,4.
225403070449829102e-01,8.378305435180664062e+00
2.840123391151428223e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.179627895355224609e+00,4.
225386083126068115e-01,8.462813377380371094e+00
2.860005998611450195e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.221881628036499023e+00,4.
225367605686187744e-01,8.547320365905761719e+00
2.884658503532409668e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.348642826080322266e+00,4.
225347638130187988e-01,8.674081802368164062e+00
2.905008792877197266e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.433149576187133789e+00,4.
225326180458068848e-01,8.758587837219238281e+00
2.929990506172180176e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.517656087875366211e+00,4.
225291609764099121e-01,8.843093872070312500e+00
2.950153279304504395e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.644414663314819336e+00,4.
225267469882965088e-01,8.927599906921386719e+00
2.974664402008056641e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.728919982910156250e+00,4.
225242137908935547e-01,9.054357528686523438e+00
2.994886708259582520e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.813424587249755859e+00,4.
225215911865234375e-01,9.138861656188964844e+00
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
3.019796800613403320e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.897928953170776367e+00,4.
225175082683563232e-01,9.223365783691406250e+00
3.040167713165283203e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.024684190750122070e+00,4.
225146770477294922e-01,9.307868957519531250e+00
3.055962896347045898e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.109186887741088867e+00,4.
225118160247802734e-01,9.392372131347656250e+00
3.070209717750549316e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.151437997817993164e+00,4.
225103557109832764e-01,9.434623718261718750e+00
3.085124301910400391e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.193689107894897461e+00,4.
225074052810668945e-01,9.519124984741210938e+00
3.109846591949462891e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.278190612792968750e+00,4.
225013554096221924e-01,9.603626251220703125e+00
3.134846806526184082e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.404940843582153320e+00,4.
224894940853118896e-01,9.730375289916992188e+00
3.155006098747253418e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.489438295364379883e+00,4.
224784374237060547e-01,9.814871788024902344e+00
3.179896092414855957e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.573933601379394531e+00,4.
224736690521240234e-01,9.899366378784179688e+00
3.204878997802734375e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.700675487518310547e+00,4.
224665462970733643e-01,1.002610778808593750e+01
3.220095682144165039e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.785168409347534180e+00,4.
224597215652465820e-01,1.006835365295410156e+01
3.245064496994018555e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.869659900665283203e+00,4.
224532842636108398e-01,1.019509029388427734e+01
3.265146279335021973e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.954150676727294922e+00,4.
224531054496765137e-01,1.027958106994628906e+01
3.285249090194702148e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.038641452789306641e+00,4.
224527180194854736e-01,1.036407184600830078e+01
3.309880089759826660e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.123131752014160156e+00,4.
224520325660705566e-01,1.044856262207031250e+01
3.334968590736389160e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.249867439270019531e+00,4.
224511086940765381e-01,1.057529830932617188e+01
3.359957408905029297e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.334357738494873047e+00,4.
224493205547332764e-01,1.065978813171386719e+01
3.384885001182556152e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.461092472076416016e+00,4.
224478304386138916e-01,1.078652286529541016e+01
3.409940910339355469e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.545581817626953125e+00,4.
224452972412109375e-01,1.087101173400878906e+01
3.434942579269409180e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.672315597534179688e+00,4.
224433898925781250e-01,1.099774551391601562e+01
3.455204510688781738e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.756803989410400391e+00,4.
224402606487274170e-01,1.108223342895507812e+01
3.479808521270751953e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.841292381286621094e+00,4.
224380254745483398e-01,1.116672134399414062e+01
3.495203304290771484e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.925779819488525391e+00,4.
224356710910797119e-01,1.125120925903320312e+01
3.519997096061706543e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.010267257690429688e+00,4.
224331974983215332e-01,1.133569622039794922e+01
3.544704198837280273e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.136997222900390625e+00,4.
224306344985961914e-01,1.146242523193359375e+01
3.564929199218750000e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.221483230590820312e+00,4.
224280118942260742e-01,1.154691123962402344e+01
3.590025281906127930e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.305968761444091797e+00,4.
224238991737365723e-01,1.163139629364013672e+01
3.614931893348693848e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.432695865631103516e+00,4.
224210977554321289e-01,1.175812339782714844e+01
3.639981102943420410e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.517179965972900391e+00,4.
224167764186859131e-01,1.184260654449462891e+01
3.660007500648498535e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.643904685974121094e+00,4.
224138557910919189e-01,1.192709064483642578e+01
3.685269403457641602e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.728387355804443359e+00,4.
224108755588531494e-01,1.205381393432617188e+01
3.709905886650085449e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.812869548797607422e+00,4.
224063456058502197e-01,1.213829612731933594e+01
3.734912109375000000e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.939591407775878906e+00,4.
224032759666442871e-01,1.226501750946044922e+01
3.755020380020141602e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.024072170257568359e+00,4.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
223971664905548096e-01,1.234949684143066406e+01
3.779920697212219238e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.108551025390625000e+00,4.
223900139331817627e-01,1.243397521972656250e+01
3.805034208297729492e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.235268115997314453e+00,4.
223874509334564209e-01,1.256069183349609375e+01
3.829948902130126953e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.656000271439552307e-02,4.
223834872245788574e-01,1.264516925811767578e+01
3.849916315078735352e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.210368350148200989e-01,4.
223747551441192627e-01,1.272964572906494141e+01
3.869911813735961914e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.055122107267379761e-01,4.
223724901676177979e-01,1.281412029266357422e+01
3.889851117134094238e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.899868190288543701e-01,4.
223702549934387207e-01,1.289859485626220703e+01
3.914969706535339355e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.166978895664215088e-01,4.
223679304122924805e-01,1.302530574798583984e+01
3.939846301078796387e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.011713504791259766e-01,4.
223642647266387939e-01,1.310977840423583984e+01
3.964713692665100098e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.278806328773498535e-01,4.
223617613315582275e-01,1.323648738861083984e+01
3.984974503517150879e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.123528718948364258e-01,4.
223591685295104980e-01,1.332095909118652344e+01
4.005282497406005859e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.968245744705200195e-01,4.
223551750183105469e-01,1.340543079376220703e+01
4.029921889305114746e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.812957406044006348e-01,4.
223524332046508789e-01,1.348990154266357422e+01
4.050219297409057617e+01,1.000507736206054688e+02,0.000000000000000000e+00,9.657663702964782715e-01,4.
223436415195465088e-01,1.357437133789062500e+01
4.065042304992675781e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.050235509872436523e+00,4.
223424494266510010e-01,1.365884017944335938e+01
4.089984798431396484e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.134703874588012695e+00,4.
223390221595764160e-01,1.374330806732177734e+01
4.114914202690124512e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.261405587196350098e+00,4.
223366379737854004e-01,1.387000942230224609e+01
4.134982490539550781e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.345872759819030762e+00,4.
223341941833496094e-01,1.395447731018066406e+01
4.159818005561828613e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.430339574813842773e+00,4.
223243594169616699e-01,1.403894329071044922e+01
4.175248789787292480e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.514804840087890625e+00,4.
223222434520721436e-01,1.412340736389160156e+01
4.199920892715454102e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.599269390106201172e+00,4.
223201274871826172e-01,1.420787239074707031e+01
4.224962615966796875e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.725965380668640137e+00,4.
223179221153259277e-01,1.433456802368164062e+01
4.249979400634765625e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.810428857803344727e+00,4.
223144352436065674e-01,1.441903114318847656e+01
4.269990086555480957e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.894891858100891113e+00,4.
223120212554931641e-01,1.450349426269531250e+01
4.294934082031250000e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.021585464477539062e+00,4.
223095476627349854e-01,1.463018703460693359e+01
4.319953989982604980e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.106047153472900391e+00,4.
223056733608245850e-01,1.471464824676513672e+01
4.340157198905944824e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.232738971710205078e+00,4.
223030507564544678e-01,1.479910945892333984e+01
4.364920997619628906e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.317199468612670898e+00,4.
223003387451171875e-01,1.492580032348632812e+01
4.385167694091796875e+01,1.000507736206054688e+02,0.000000000000000000e+00,2.401659011840820312e+00,4.
222915470600128174e-01,1.501025867462158203e+01
4.400202012062072754e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.486117362976074219e+00,4.
222892820835113525e-01,1.505248832702636719e+01
4.425108814239501953e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.570574998855590820e+00,4.
222870171070098877e-01,1.517917442321777344e+01
4.449910092353820801e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.655032396316528320e+00,4.
222834706306457520e-01,1.526363182067871094e+01
4.470177984237670898e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.781717300415039062e+00,4.
222810268402099609e-01,1.534808826446533203e+01
4.489857292175292969e+01,1.000507736206054688e+02,0.000000000000000000e+00,2.823945522308349609e+00,4.
222724735736846924e-01,1.543254375457763672e+01
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
4.509893202781677246e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.908400297164916992e+00,4.
222703874111175537e-01,1.551699829101562500e+01
4.534851193428039551e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.035081386566162109e+00,4.
222683310508728027e-01,1.564367866516113281e+01
4.554912614822387695e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.119534969329833984e+00,4.
222661554813385010e-01,1.572813224792480469e+01
4.574943208694458008e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.203988075256347656e+00,4.
222639203071594238e-01,1.581258487701416016e+01
4.595231318473815918e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.288440942764282227e+00,4.
222603738307952881e-01,1.589703750610351562e+01
4.619864010810852051e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.372893095016479492e+00,4.
222579300403594971e-01,1.598148918151855469e+01
4.639876985549926758e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.457344770431518555e+00,4.
222554266452789307e-01,1.606594085693359375e+01
4.659940886497497559e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.541795969009399414e+00,4.
222528636455535889e-01,1.615039253234863281e+01
4.685108399391174316e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.668471813201904297e+00,4.
222471714019775391e-01,1.627706718444824219e+01
4.709859085083007812e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.752920866012573242e+00,4.
222346246242523193e-01,1.636151504516601562e+01
4.725243806838989258e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.837368011474609375e+00,4.
222329556941986084e-01,1.644596290588378906e+01
4.749926805496215820e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.921814680099487305e+00,4.
222312569618225098e-01,1.653040885925292969e+01
4.769963693618774414e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.006261348724365234e+00,4.
222294390201568604e-01,1.661485481262207031e+01
4.794874286651611328e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.132929801940917969e+00,4.
222275018692016602e-01,1.674152374267578125e+01
4.815018010139465332e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.217375278472900391e+00,4.
222254455089569092e-01,1.682596778869628906e+01
4.839946222305297852e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.301820278167724609e+00,4.
222221970558166504e-01,1.691041374206542969e+01
4.859988808631896973e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.386264801025390625e+00,4.
222199022769927979e-01,1.699485778808593750e+01
4.880130195617675781e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.512930870056152344e+00,4.
222175478935241699e-01,1.707930183410644531e+01
4.904934597015380859e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.597374439239501953e+00,4.
222151339054107666e-01,1.720596694946289062e+01
4.924920201301574707e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.681817054748535156e+00,4.
222095906734466553e-01,1.729040908813476562e+01
4.944955110549926758e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.766258716583251953e+00,4.
222042560577392578e-01,1.737484931945800781e+01
4.964911794662475586e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.850699424743652344e+00,4.
222022294998168945e-01,1.745928955078125000e+01
4.984981513023376465e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.935139656066894531e+00,4.
222001135349273682e-01,1.754372978210449219e+01
5.009975910186767578e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.019579410552978516e+00,4.
221967756748199463e-01,1.762817001342773438e+01
5.034969592094421387e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.146238803863525391e+00,4.
221944510936737061e-01,1.775482940673828125e+01
5.055281686782836914e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.230677604675292969e+00,4.
221920669078826904e-01,1.783926773071289062e+01
5.065002799034118652e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.272896766662597656e+00,4.
221908450126647949e-01,1.788148689270019531e+01
5.085559797286987305e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.357334613800048828e+00,4.
221870899200439453e-01,1.796592330932617188e+01
5.110019087791442871e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.441772460937500000e+00,4.
221844971179962158e-01,1.805036163330078125e+01
5.134880685806274414e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.568427562713623047e+00,4.
221818745136260986e-01,1.817701721191406250e+01
5.150221586227416992e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.652863979339599609e+00,4.
221792221069335938e-01,1.821923446655273438e+01
5.174962401390075684e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.737299442291259766e+00,4.
221734702587127686e-01,1.834588813781738281e+01
5.194984388351440430e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.821733951568603516e+00,4.
221679568290710449e-01,1.843032073974609375e+01
5.220009899139404297e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.906167507171630859e+00,4.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
221616089344024658e-01,1.851475524902343750e+01
5.235212492942810059e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.990600109100341797e+00,4.
221565425395965576e-01,1.859918594360351562e+01
5.259889101982116699e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.075031280517578125e+00,4.
221517145633697510e-01,1.868361854553222656e+01
5.279849100112915039e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.159461975097656250e+00,4.
221439957618713379e-01,1.876804733276367188e+01
5.304970002174377441e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.920062048360705376e-03,4.
221367537975311279e-01,1.889468955993652344e+01
5.325066113471984863e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.734706789255142212e-02,4.
221330285072326660e-01,1.897911643981933594e+01
5.349978995323181152e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.717736423015594482e-01,4.
221321046352386475e-01,1.906354331970214844e+01
5.370025396347045898e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.562001049518585205e-01,4.
221312403678894043e-01,1.914796829223632812e+01
5.394946789741516113e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.828394711017608643e-01,4.
221301674842834473e-01,1.927460861206054688e+01
5.415299606323242188e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.672654569149017334e-01,4.
221282303333282471e-01,1.935903358459472656e+01
5.440067982673645020e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.516911745071411133e-01,4.
221267402172088623e-01,1.944346046447753906e+01
5.465024614334106445e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.783291697502136230e-01,4.
221251010894775391e-01,1.957009696960449219e+01
5.489927911758422852e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.627541422843933105e-01,4.
221224188804626465e-01,1.965452194213867188e+01
5.510268688201904297e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.893908262252807617e-01,4.
221204519271850586e-01,1.973894691467285156e+01
5.534945106506347656e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.738148450851440430e-01,4.
221183955669403076e-01,1.986558341979980469e+01
5.555120611190795898e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.058238148689270020e+00,4.
221101999282836914e-01,1.995000457763671875e+01
5.575194215774536133e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.142660021781921387e+00,4.
221015572547912598e-01,2.003442573547363281e+01
5.599916386604309082e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.227080702781677246e+00,4.
221002459526062012e-01,2.011884689331054688e+01
5.625047087669372559e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.353710770606994629e+00,4.
220988750457763672e-01,2.024547576904296875e+01
5.645028614997863770e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.438130497932434082e+00,4.
220973551273345947e-01,2.032989692687988281e+01
5.669954514503479004e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.522549867630004883e+00,4.
220887720584869385e-01,2.041431427001953125e+01
5.689996695518493652e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.606968045234680176e+00,4.
220874309539794922e-01,2.049873352050781250e+01
5.714969491958618164e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.733594298362731934e+00,4.
220860302448272705e-01,2.062535858154296875e+01
5.730261611938476562e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.818011403083801270e+00,4.
220844805240631104e-01,2.066756820678710938e+01
5.755396103858947754e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.902428150177001953e+00,4.
220828115940093994e-01,2.079419326782226562e+01
5.774982714653015137e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.986844658851623535e+00,4.
220809936523437500e-01,2.087860870361328125e+01
5.799964904785156250e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.071260690689086914e+00,4.
220780730247497559e-01,2.096302413940429688e+01
5.824975514411926270e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.197884321212768555e+00,4.
220760166645050049e-01,2.108964729309082031e+01
5.849993705749511719e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.282299280166625977e+00,4.
220727384090423584e-01,2.117406272888183594e+01
5.875153994560241699e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.408921241760253906e+00,4.
220704734325408936e-01,2.130068397521972656e+01
5.899951505661010742e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.493335008621215820e+00,4.
220638573169708252e-01,2.138509750366210938e+01
5.925131106376647949e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.619954347610473633e+00,4.
220586419105529785e-01,2.151171684265136719e+01
5.949936413764953613e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.704365968704223633e+00,4.
220557510852813721e-01,2.159612655639648438e+01
5.969947695732116699e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.788777112960815430e+00,4.
220536947250366211e-01,2.168053817749023438e+01
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
5.994977998733520508e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.915393352508544922e+00,4.
220515787601470947e-01,2.180715370178222656e+01
6.015106296539306641e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.999803543090820312e+00,4.
220493733882904053e-01,2.189156341552734375e+01
6.035185408592224121e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.084213256835937500e+00,4.
220459163188934326e-01,2.197597312927246094e+01
6.055048489570617676e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.168622493743896484e+00,4.
220447540283203125e-01,2.206038284301757812e+01
6.069875001907348633e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.210827112197875977e+00,4.
220423400402069092e-01,2.210258674621582031e+01
6.090234589576721191e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.337439775466918945e+00,4.
220398962497711182e-01,2.218699645996093750e+01
6.110020613670349121e+01,1.000507736206054688e+02,0.000000000000000000e+00,3.379643678665161133e+00,4.
220283031463623047e-01,2.227140235900878906e+01
6.134936809539794922e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.506252527236938477e+00,4.
220235049724578857e-01,2.239801025390625000e+01
6.155063915252685547e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.590657234191894531e+00,4.
220220744609832764e-01,2.248241615295410156e+01
6.179990911483764648e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.675061464309692383e+00,4.
220196902751922607e-01,2.256682014465332031e+01
6.204691982269287109e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.801667451858520508e+00,4.
220179617404937744e-01,2.269342422485351562e+01
6.220194602012634277e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.886070966720581055e+00,4.
220161139965057373e-01,2.273562622070312500e+01
6.239958405494689941e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.928272485733032227e+00,4.
220142066478729248e-01,2.282003021240234375e+01
6.264976787567138672e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.054876804351806641e+00,4.
220121800899505615e-01,2.294663429260253906e+01
6.290001106262207031e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.139278888702392578e+00,4.
220089912414550781e-01,2.303103637695312500e+01
6.315047597885131836e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.265882015228271484e+00,4.
220067560672760010e-01,2.315763854980468750e+01
6.339922618865966797e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.350283145904541016e+00,4.
220033288002014160e-01,2.324203872680664062e+01
6.365032696723937988e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.476883888244628906e+00,4.
220009446144104004e-01,2.336864089965820312e+01
6.384938216209411621e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.561284065246582031e+00,4.
219985306262969971e-01,2.345303916931152344e+01
6.410028791427612305e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.645683765411376953e+00,4.
219948053359985352e-01,2.353743934631347656e+01
6.435086989402770996e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.772281646728515625e+00,4.
219862520694732666e-01,2.366403579711914062e+01
6.460192418098449707e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.856678962707519531e+00,4.
219831824302673340e-01,2.374843406677246094e+01
6.485102319717407227e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.983273983001708984e+00,4.
219811260700225830e-01,2.387502861022949219e+01
6.500204801559448242e+01,1.000507736206054688e+02,0.000000000000000000e+00,5.067669868469238281e+00,4.
219729602336883545e-01,2.391722488403320312e+01
6.525075507164001465e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.152064323425292969e+00,4.
219712615013122559e-01,2.404381752014160156e+01
6.549871516227722168e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.236458301544189453e+00,4.
219686686992645264e-01,2.412821197509765625e+01
6.570206093788146973e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.363049030303955078e+00,4.
219637811183929443e-01,2.421260452270507812e+01
6.590076398849487305e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.405245304107666016e+00,4.
219590425491333008e-01,2.429699707031250000e+01
6.614988994598388672e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.531833171844482422e+00,4.
219576120376586914e-01,2.442358398437500000e+01
6.635331702232360840e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.616224765777587891e+00,4.
219552278518676758e-01,2.450797653198242188e+01
6.659991502761840820e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.700615882873535156e+00,4.
219534993171691895e-01,2.459236717224121094e+01
6.685087513923645020e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.827201843261718750e+00,4.
219516515731811523e-01,2.471895217895507812e+01
6.705002808570861816e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.911592006683349609e+00,4.
219497442245483398e-01,2.480334281921386719e+01
6.725059890747070312e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.995981693267822266e+00,4.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
219477176666259766e-01,2.488773155212402344e+01
6.744945883750915527e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.080371379852294922e+00,4.
219456315040588379e-01,2.497212219238281250e+01
6.769993495941162109e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.164760112762451172e+00,4.
219423532485961914e-01,2.505651092529296875e+01
6.794939112663269043e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.157648146152496338e-03,4.
219400882720947266e-01,2.518309211730957031e+01
6.814976310729980469e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.254555404186248779e-02,4.
219377934932708740e-01,2.526748085021972656e+01
6.834917807579040527e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.769329905509948730e-01,4.
219354093074798584e-01,2.535186767578125000e+01
6.855005002021789551e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.613199651241302490e-01,4.
219330251216888428e-01,2.543625450134277344e+01
6.880047988891601562e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.457064330577850342e-01,4.
219293296337127686e-01,2.552064132690429688e+01
6.904995703697204590e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.722852408885955811e-01,4.
219268262386322021e-01,2.564721870422363281e+01
6.930230498313903809e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.566704869270324707e-01,4.
219230115413665771e-01,2.573160362243652344e+01
6.955002593994140625e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.832473874092102051e-01,4.
219204485416412354e-01,2.585818099975585938e+01
6.975017595291137695e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.676313519477844238e-01,4.
219178557395935059e-01,2.594256401062011719e+01
6.995332813262939453e+01,1.000507736206054688e+02,0.000000000000000000e+00,8.520144820213317871e-01,4.
219081103801727295e-01,2.602694702148437500e+01
7.014967083930969238e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.363961815834045410e-01,4.
219070971012115479e-01,2.611132812500000000e+01
7.040071082115173340e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.020777463912963867e+00,4.
219039380550384521e-01,2.619570922851562500e+01
7.056108117103576660e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.147348642349243164e+00,4.
219017624855041504e-01,2.628009033203125000e+01
7.075209808349609375e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.189538836479187012e+00,4.
219006597995758057e-01,2.636446952819824219e+01
7.100262594223022461e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.273918867111206055e+00,4.
218972921371459961e-01,2.644885063171386719e+01
7.115324592590332031e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.358298420906066895e+00,4.
218949675559997559e-01,2.653322982788085938e+01
7.135041999816894531e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.442677617073059082e+00,4.
218938052654266357e-01,2.661760902404785156e+01
7.154979896545410156e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.527055978775024414e+00,4.
218854010105133057e-01,2.670198631286621094e+01
7.175193905830383301e+01,1.000507736206054688e+02,0.000000000000000000e+00,1.611432552337646484e+00,4.
218774437904357910e-01,2.678636169433593750e+01
7.199932098388671875e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.695808053016662598e+00,4.
218753874301910400e-01,2.687073707580566406e+01
7.224991798400878906e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.822370648384094238e+00,4.
218739569187164307e-01,2.699729919433593750e+01
7.245277094841003418e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.906745314598083496e+00,4.
218715429306030273e-01,2.708167457580566406e+01
7.264955210685729980e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.991119742393493652e+00,4.
218707084655761719e-01,2.716604804992675781e+01
7.285032701492309570e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.075493812561035156e+00,4.
218689501285552979e-01,2.725042152404785156e+01
7.309962010383605957e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.159867525100708008e+00,4.
218661189079284668e-01,2.733479499816894531e+01
7.325338482856750488e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.244240760803222656e+00,4.
218641519546508789e-01,2.741916847229003906e+01
7.350095510482788086e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.328613758087158203e+00,4.
218620955944061279e-01,2.750354194641113281e+01
7.375079703330993652e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.455172300338745117e+00,4.
218599796295166016e-01,2.763010025024414062e+01
7.399859619140625000e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.539544343948364258e+00,4.
218567013740539551e-01,2.771447181701660156e+01
7.415325784683227539e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.623915672302246094e+00,4.
218544363975524902e-01,2.779884338378906250e+01
7.440037798881530762e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.708286762237548828e+00,4.
218521416187286377e-01,2.788321304321289062e+01
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
7.465076398849487305e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.834842443466186523e+00,4.
218467473983764648e-01,2.800976753234863281e+01
7.490125298500061035e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.919211387634277344e+00,4.
218406677246093750e-01,2.809413719177246094e+01
7.515054988861083984e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.045763492584228516e+00,4.
218387901782989502e-01,2.822068786621093750e+01
7.540041899681091309e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.130131244659423828e+00,4.
218358695507049561e-01,2.830505561828613281e+01
7.565081596374511719e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.256681919097900391e+00,4.
218338131904602051e-01,2.843160629272460938e+01
7.585109305381774902e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.341048479080200195e+00,4.
218316972255706787e-01,2.851597213745117188e+01
7.610126304626464844e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.425414800643920898e+00,4.
218223989009857178e-01,2.860033798217773438e+01
7.630262494087219238e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.551961898803710938e+00,4.
218206703662872314e-01,2.868470382690429688e+01
7.650165820121765137e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.636325836181640625e+00,4.
218189418315887451e-01,2.876906776428222656e+01
7.670005702972412109e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.678507804870605469e+00,4.
218171536922454834e-01,2.885343170166015625e+01
7.694942498207092285e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.805052995681762695e+00,4.
218152761459350586e-01,2.897997474670410156e+01
7.710221004486083984e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.889415979385375977e+00,4.
218133091926574707e-01,2.902215766906738281e+01
7.735039520263671875e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.973778486251831055e+00,4.
218112826347351074e-01,2.914870071411132812e+01
7.755063080787658691e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.058140754699707031e+00,4.
218091964721679688e-01,2.923306274414062500e+01
7.779995203018188477e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.142502307891845703e+00,4.
218059480190277100e-01,2.931742477416992188e+01
7.805085206031799316e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.269043922424316406e+00,4.
218037426471710205e-01,2.944396591186523438e+01
7.829933404922485352e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.353404521942138672e+00,4.
217884540557861328e-01,2.952832412719726562e+01
7.855062818527221680e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.479941368103027344e+00,4.
217872321605682373e-01,2.965486145019531250e+01
7.879998421669006348e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.564298629760742188e+00,4.
217852950096130371e-01,2.973921775817871094e+01
7.905053997039794922e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.690834045410156250e+00,4.
217838644981384277e-01,2.986575317382812500e+01
7.925098800659179688e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.775190830230712891e+00,4.
217823147773742676e-01,2.995010948181152344e+01
7.945049595832824707e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.859547138214111328e+00,4.
217806756496429443e-01,3.003446578979492188e+01
7.970097208023071289e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.943903446197509766e+00,4.
217780232429504395e-01,3.011882209777832031e+01
7.995160698890686035e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.070436954498291016e+00,4.
217761456966400146e-01,3.024535560607910156e+01
8.020243191719055176e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.154791831970214844e+00,4.
217701554298400879e-01,3.032971000671386719e+01
8.045019698143005371e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.281322956085205078e+00,4.
217653572559356689e-01,3.045624160766601562e+01
8.056172204017639160e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.365675926208496094e+00,4.
217638373374938965e-01,3.049841690063476562e+01
8.070224213600158691e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.407852172851562500e+00,4.
217630326747894287e-01,3.054059410095214844e+01
8.090122914314270020e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.450028896331787109e+00,4.
217613637447357178e-01,3.062494659423828125e+01
8.109923195838928223e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.534380912780761719e+00,4.
217596054077148438e-01,3.070929908752441406e+01
8.130354189872741699e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.660908699035644531e+00,4.
217577576637268066e-01,3.079364967346191406e+01
8.155168700218200684e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.745260238647460938e+00,4.
217558503150939941e-01,3.092017745971679688e+01
8.175066208839416504e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.829611301422119141e+00,4.
217538535594940186e-01,3.100452804565429688e+01
8.200163888931274414e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.913961887359619141e+00,4.
```

```
217507839202880859e-01,3.108887863159179688e+01
8.225056314468383789e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.040487289428710938e+00,4.
217486381530761719e-01,3.121540260314941406e+01
8.250143098831176758e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.124836921691894531e+00,4.
217393100261688232e-01,3.129975318908691406e+01
8.275047492980957031e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.251358985900878906e+00,4.
217375814914703369e-01,3.142627334594726562e+01
8.300008392333984375e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.252119898796081543e-02,4.
217349886894226074e-01,3.151062202453613281e+01
8.325002384185791016e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.790416836738586426e-01,4.
217331409454345703e-01,3.163714218139648438e+01
8.345030593872070312e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.633882164955139160e-01,4.
217312335968017578e-01,3.172148704528808594e+01
8.369996786117553711e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.477343618869781494e-01,4.
217282533645629883e-01,3.180583381652832031e+01
8.395029187202453613e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.742528498172760010e-01,4.
217261970043182373e-01,3.193235206604003906e+01
8.415342903137207031e+01,1.000507736206054688e+02,0.000000000000000000e+00,5.585976839065551758e-01,4.
217171967029571533e-01,3.201669692993164062e+01
8.440286111831665039e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.429411768913269043e-01,4.
217155873775482178e-01,3.210103988647460938e+01
8.465327501296997070e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.694558501243591309e-01,4.
217139184474945068e-01,3.222755432128906250e+01
8.490037703514099121e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.537985682487487793e-01,4.
217112362384796143e-01,3.231189727783203125e+01
8.515139412879943848e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.803119301795959473e-01,4.
217093586921691895e-01,3.243840789794921875e+01
8.535338997840881348e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.064653754234313965e+00,4.
217064380645751953e-01,3.252275085449218750e+01
8.555128192901611328e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.148995161056518555e+00,4.
217054247856140137e-01,3.260709381103515625e+01
8.579951691627502441e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.233336091041564941e+00,4.
217023253440856934e-01,3.269143295288085938e+01
8.600350785255432129e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.359846830368041992e+00,4.
217002093791961670e-01,3.277577209472656250e+01
8.620170211791992188e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.444186687469482422e+00,4.
216949939727783203e-01,3.286011505126953125e+01
8.640249109268188477e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.528525352478027344e+00,4.
216900169849395752e-01,3.294445037841796875e+01
8.665033388137817383e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.612863183021545410e+00,4.
216883480548858643e-01,3.307095718383789062e+01
8.690126585960388184e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.697200775146484375e+00,4.
216826558113098145e-01,3.315529632568359375e+01
8.714800405502319336e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.823705673217773438e+00,4.
216780364513397217e-01,3.328179931640625000e+01
8.735160589218139648e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.908041119575500488e+00,4.
216767251491546631e-01,3.336613464355468750e+01
8.754781889915466309e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.992376446723937988e+00,4.
216752946376800537e-01,3.345046997070312500e+01
8.775170397758483887e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.076711416244506836e+00,4.
216737449169158936e-01,3.353480529785156250e+01
8.795299983024597168e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.161046028137207031e+00,4.
216712713241577148e-01,3.361914062500000000e+01
8.815137887001037598e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.245380401611328125e+00,4.
216704070568084717e-01,3.370347213745117188e+01
8.840001201629638672e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.329714536666870117e+00,4.
216677248477935791e-01,3.378780746459960938e+01
8.865057086944580078e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.456214666366577148e+00,4.
216658174991607666e-01,3.391430664062500000e+01
8.890135693550109863e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.540547847747802734e+00,4.
216628670692443848e-01,3.399864196777343750e+01
8.915048384666442871e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.667046785354614258e+00,4.
216608405113220215e-01,3.412514114379882812e+01
8.940175294876098633e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.751378774642944336e+00,4.
216576814651489258e-01,3.420947265625000000e+01
8.965016388893127441e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.877876043319702148e+00,4.
216555356979370117e-01,3.433596801757812500e+01
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
8.985077404975891113e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.962207078933715820e+00,4.
216533601284027100e-01,3.442029953002929688e+01
9.005221295356750488e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.046537637710571289e+00,4.
216511249542236328e-01,3.450463104248046875e+01
9.030098891258239746e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.130867719650268555e+00,4.
216477572917938232e-01,3.458895874023437500e+01
9.050286316871643066e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.257362127304077148e+00,4.
216424226760864258e-01,3.467329025268554688e+01
9.065054297447204590e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.299526214599609375e+00,4.
216382801532745361e-01,3.475761795043945312e+01
9.085070896148681641e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.383853912353515625e+00,4.
216334223747253418e-01,3.484194183349609375e+01
9.110279083251953125e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.468180179595947266e+00,4.
216282367706298828e-01,3.492626953125000000e+01
9.129986000061035156e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.552505731582641602e+00,4.
216269254684448242e-01,3.501059341430664062e+01
9.155163908004760742e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.678993940353393555e+00,4.
216255247592926025e-01,3.513708114624023438e+01
9.175329518318176270e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.763319015502929688e+00,4.
216232299804687500e-01,3.522140884399414062e+01
9.200089120864868164e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.847643613815307617e+00,4.
216215908527374268e-01,3.530573272705078125e+01
9.220338296890258789e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.974130153656005859e+00,4.
216198921203613281e-01,3.539005661010742188e+01
9.245011496543884277e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.058454036712646484e+00,4.
216181039810180664e-01,3.551654052734375000e+01
9.270027208328247070e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.142777442932128906e+00,4.
216153025627136230e-01,3.560086441040039062e+01
9.295018696784973145e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.269262313842773438e+00,4.
216133654117584229e-01,3.572734832763671875e+01
9.315019583702087402e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.353584766387939453e+00,4.
216113686561584473e-01,3.581167221069335938e+01
9.335372495651245117e+01,1.000507736206054688e+02,0.000000000000000000e+00,4.437906742095947266e+00,4.
216024577617645264e-01,3.589599227905273438e+01
9.360081791877746582e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.522227287292480469e+00,4.
216009378433227539e-01,3.598031234741210938e+01
9.385064506530761719e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.648707389831542969e+00,4.
215993285179138184e-01,3.610679244995117188e+01
9.409960293769836426e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.733027458190917969e+00,4.
215997755527496338e-01,3.619111251831054688e+01
9.435192084312438965e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.859507083892822266e+00,4.
215947091579437256e-01,3.631759262084960938e+01
9.460095214843750000e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.943825721740722656e+00,4.
215919375419616699e-01,3.640191268920898438e+01
9.480130290985107422e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.028144359588623047e+00,4.
215869903564453125e-01,3.648622894287109375e+01
9.500015902519226074e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.112462043762207031e+00,4.
215822219848632812e-01,3.657054519653320312e+01
9.525161504745483398e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.238936424255371094e+00,4.
215808212757110596e-01,3.669702148437500000e+01
9.550080299377441406e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.323252677917480469e+00,4.
215784966945648193e-01,3.678133773803710938e+01
9.570224499702453613e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.449726104736328125e+00,4.
215768575668334961e-01,3.686565399169921875e+01
9.595214319229125977e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.534041404724121094e+00,4.
215751588344573975e-01,3.699212646484375000e+01
9.615060710906982422e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.618356704711914062e+00,4.
215733706951141357e-01,3.707643890380859375e+01
9.635363602638244629e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.702671051025390625e+00,4.
215705990791320801e-01,3.716075515747070312e+01
9.660038614273071289e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.786985397338867188e+00,4.
215686619281768799e-01,3.724506759643554688e+01
9.685164809226989746e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.913455963134765625e+00,4.
215666949748992920e-01,3.737154006958007812e+01
9.710094213485717773e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.997769355773925781e+00,4.
215636551380157471e-01,3.745585250854492188e+01
9.735049295425415039e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.124238014221191406e+00,4.
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
215615689754486084e-01,3.758232116699218750e+01
9.755405116081237793e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.208550453186035156e+00,4.
215583801269531250e-01,3.766663360595703125e+01
9.780400300025939941e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.676922112703323364e-03,4.
215562343597412109e-01,3.775094604492187500e+01
9.805192708969116211e+01,1.000761566162109375e+02,0.000000000000000000e+00,1.361437886953353882e-01,4.
215540289878845215e-01,3.787741088867187500e+01
9.830088710784912109e+01,1.000761566162109375e+02,0.000000000000000000e+00,2.204544842243194580e-01,4.
215507209300994873e-01,3.796172332763671875e+01
9.855082798004150391e+01,1.000761566162109375e+02,0.000000000000000000e+00,3.469197154045104980e-01,4.
215484857559204102e-01,3.808818817138671875e+01
9.880070304870605469e+01,1.000761566162109375e+02,0.000000000000000000e+00,4.312292933464050293e-01,4.
215451180934906006e-01,3.817249679565429688e+01
9.905079507827758789e+01,1.000761566162109375e+02,0.000000000000000000e+00,5.576928257942199707e-01,4.
215428233146667480e-01,3.829895782470703125e+01
9.925320196151733398e+01,1.000761566162109375e+02,0.000000000000000000e+00,6.420012712478637695e-01,4.
215393960475921631e-01,3.838326644897460938e+01
9.950030207633972168e+01,1.000761566162109375e+02,0.000000000000000000e+00,7.263092994689941406e-01,4.
215371012687683105e-01,3.846757507324218750e+01
9.970344185829162598e+01,1.000761566162109375e+02,0.000000000000000000e+00,8.527703881263732910e-01,4.
215317368507385254e-01,3.855188369750976562e+01
9.994999599456787109e+01,1.000761566162109375e+02,0.000000000000000000e+00,9.370763301849365234e-01,4.
215266704559326172e-01,3.867834091186523438e+01
1.001524589061737061e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.021381616592407227e+00,4.
215249121189117432e-01,3.876264572143554688e+01
1.004007840156555176e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.105686426162719727e+00,4.
215222299098968506e-01,3.884695053100585938e+01
1.005624928474426270e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.189990997314453125e+00,4.
215203523635864258e-01,3.893125534057617188e+01
1.007503650188446045e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.274295210838317871e+00,4.
215194284915924072e-01,3.901556015014648438e+01
1.010006148815155029e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.358598947525024414e+00,4.
215165078639984131e-01,3.909986114501953125e+01
1.012509579658508301e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.485053896903991699e+00,4.
215145409107208252e-01,3.922631835937500000e+01
1.015013530254364014e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.569356679916381836e+00,4.
215115010738372803e-01,3.931061935424804688e+01
1.017511680126190186e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.695810198783874512e+00,4.
215094447135925293e-01,3.943707275390625000e+01
1.019512350559234619e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.780111908912658691e+00,4.
215073585510253906e-01,3.952137374877929688e+01
1.022011730670928955e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.864413380622863770e+00,4.
215041697025299072e-01,3.960567474365234375e+01
1.024504840373992920e+02,1.000761566162109375e+02,0.000000000000000000e+00,1.990864634513854980e+00,4.
214989840984344482e-01,3.973212814331054688e+01
1.027011210918426514e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.075164079666137695e+00,4.
214932322502136230e-01,3.981642532348632812e+01
1.029503910541534424e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.201611995697021484e+00,4.
214916229248046875e-01,3.994287109375000000e+01
1.032014091014862061e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.285910129547119141e+00,4.
214890599250793457e-01,4.002717208862304688e+01
1.034023909568786621e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.412356853485107422e+00,4.
214872717857360840e-01,4.011146926879882812e+01
1.036520409584045410e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.496654272079467773e+00,4.
214854538440704346e-01,4.023791503906250000e+01
1.038997061252593994e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.580951213836669922e+00,4.
214826226234436035e-01,4.032221221923828125e+01
1.041034610271453857e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.707396030426025391e+00,4.
214807152748107910e-01,4.040650939941406250e+01
1.043501460552215576e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.791692018508911133e+00,4.
214787483215332031e-01,4.053295135498046875e+01
1.045058941841125488e+02,1.000507736206054688e+02,0.000000000000000000e+00,2.875987529754638672e+00,4.
214707016944885254e-01,4.057509994506835938e+01
1.047518038749694824e+02,1.000761566162109375e+02,0.000000000000000000e+00,2.960281610488891602e+00,4.
214691519737243652e-01,4.070154190063476562e+01
1.049521260261535645e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.044575214385986328e+00,4.
214676618576049805e-01,4.078583526611328125e+01
```

File - /Users/espen/PycharmProjects/thruster-simulator/Python/logs/log_azi_step_100ccw.csv

```
1.051478960514068604e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.128868818283081055e+00,4.
214660823345184326e-01,4.087012863159179688e+01
1.053519940376281738e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.213161945343017578e+00,4.
214644432067871094e-01,4.095442199707031250e+01
1.056010220050811768e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.297454595565795898e+00,4.
214618802070617676e-01,4.103871154785156250e+01
1.057539901733398438e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.381747007369995117e+00,4.
214600920677185059e-01,4.112300491333007812e+01
1.060005691051483154e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.466039180755615234e+00,4.
214582443237304688e-01,4.120729827880859375e+01
1.062033238410949707e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.592476606369018555e+00,4.
214563667774200439e-01,4.129158782958984375e+01
1.064503018856048584e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.676767826080322266e+00,4.
214544594287872314e-01,4.141802597045898438e+01
1.066520271301269531e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.761058568954467773e+00,4.
214524924755096436e-01,4.150231552124023438e+01
1.068512029647827148e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.845349073410034180e+00,4.
214504957199096680e-01,4.158660507202148438e+01
1.070522658824920654e+02,1.000761566162109375e+02,0.000000000000000000e+00,3.929639101028442383e+00,4.
214484691619873047e-01,4.167089462280273438e+01
1.073005430698394775e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.013928890228271484e+00,4.
214453995227813721e-01,4.175518417358398438e+01
1.075513279438018799e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.140362262725830078e+00,4.
214433133602142334e-01,4.188161849975585938e+01
1.077028689384460449e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.224650859832763672e+00,4.
214411973953247070e-01,4.192376327514648438e+01
1.079526991844177246e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.308938980102539062e+00,4.
214390814304351807e-01,4.205019378662109375e+01
1.082004179954528809e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.393226623535156250e+00,4.
214358627796173096e-01,4.213448333740234375e+01
1.084041700363159180e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.519657611846923828e+00,4.
214336872100830078e-01,4.221876907348632812e+01
1.086508228778839111e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.603944301605224609e+00,4.
214315414428710938e-01,4.234519958496093750e+01
1.089008970260620117e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.688230037689208984e+00,4.
214222133159637451e-01,4.242948532104492188e+01
1.091516239643096924e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.814657211303710938e+00,4.
214205145835876465e-01,4.255591201782226562e+01
1.093521819114685059e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.898941040039062500e+00,4.
214188754558563232e-01,4.264019393920898438e+01
1.095512571334838867e+02,1.000761566162109375e+02,0.000000000000000000e+00,4.983224868774414062e+00,4.
214172065258026123e-01,4.272447967529296875e+01
1.098019130229949951e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.067508220672607422e+00,4.
214145541191101074e-01,4.280876159667968750e+01
1.100510120391845703e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.193932533264160156e+00,4.
214127659797668457e-01,4.293518447875976562e+01
1.103017878532409668e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.278214931488037109e+00,4.
214039444923400879e-01,4.301946640014648438e+01
1.105503261089324951e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.404636859893798828e+00,4.
214025437831878662e-01,4.314588928222656250e+01
1.106486780643463135e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.446776866912841797e+00,4.
214018583297729492e-01,4.318803024291992188e+01
1.108533251285552979e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.531057357788085938e+00,4.
214004278182983398e-01,4.327230834960937500e+01
1.111023449897766113e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.615337371826171875e+00,4.
213981926441192627e-01,4.335659027099609375e+01
1.113005759716033936e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.699616909027099609e+00,4.
213935434818267822e-01,4.344086837768554688e+01
1.115034728050231934e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.826035022735595703e+00,4.
213891327381134033e-01,4.352514648437500000e+01
1.117505111694335938e+02,1.000761566162109375e+02,0.000000000000000000e+00,5.910312652587890625e+00,4.
213880002498626709e-01,4.365156173706054688e+01
```

Improving the testing of the Mcon propulsion control system

NTNU

Norwegian University of
Science and Technology