Description of simplified DP simulator for Gunnerus

1 INTRODUCTION

An FMU based simulator for recreating DP operations for NTNU's research vessel Gunnerus has been created as part of the Open Simulation Platform (OSP) project. The simulations have been run using the Core Simulation Environment (CSE) developed as part of OSP and two different scenarios have been configured to demonstrate distributed and centralized simulations.

The purpose of this document is to provide documentation for each of the FMUs comprising the system as well as instructions and requirements for running the distributed and centralized simulations.

2 FMUS

The simulations are FMU based with each FMU representing an integral part of Gunnerus' DP system. Most of the FMUS are based on Simulink models developed by NTNU and have been exported from Simulink using Modelon's FMU exporter. Efforts have been made to preserve model behavior and functionality, but due to limitations with the FMU exporter some parts of the original model have been removed or simplified.

This section will briefly introduce the FMUs making up the simulation. The role of each component will be explained as well as its interface and parameters. Each section will also contain a list of tunables, which are the subset of FMU parameters that can be changed without compromising the stability of the model.

2.1 BOXREFERENCE.FMU

Error! Reference source not found. shows the Simulink implementation of the box reference FMU. The responsibility of the subsystem is to generate a box reference signal for the DP control system. The reference is generated by several step signals which are activated at different points in time in the simulation. The output is the desired vessel position in the NED frame.

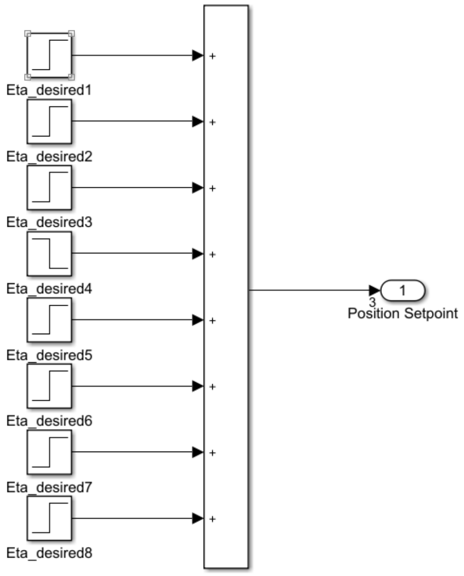

Figure 1 Box Reference Interface

The resulting box plot is shown in **Error! Reference source not found.**. North position is set to 40m at 500s and to 0m at 2000s, east to -40m at 1000s and 0 at 2500s and heading to -0.8 rad at 1500s and 0 at 2500s.

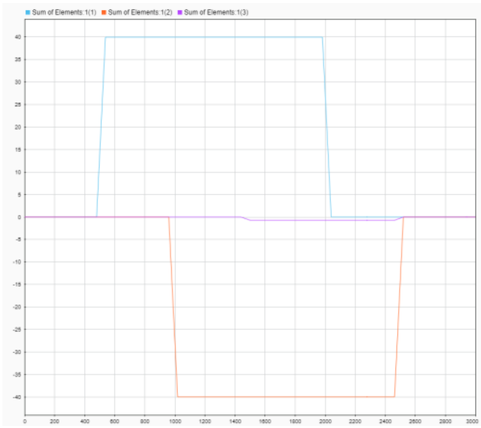

Figure 2 Box Plot

Name	Unit/Dimension	Causality
Position Setpoint	[m, m, rad]	Output

2.2 ControlSystemCommunication.fmu

The control system communication FMU is component that transmits data to and receives data from the control system. Its input and output ports are described Table 1. In this context, it is meant to represent network communication with a control system that runs outside of the FMU environment.

Transmitted data is encoded as JSON and sent to a port and address specified in a configuration file, located in the FMUs resource folder. Each JSON message consists of a signal name, type, and value field. The name and type are obtained from the modelDescription.xml file.

Received data is configured by specifying variables with ouput causality in the modelDescription.xml file. The FMU listens for messages matching the type and name of each output variable. A JSON message and corresponding entry in the model description are shown in Listing 1 and Listing 2.

```
{
name : " Po s i t i on "
type : "Real "
value : 20. 0
}
```

Listing 1 Example of JSON Message

```
<ScalarVariable name="Position " valueReference="1" causality = " output">
<Real/>
</ScalarVariable>
```

Listing 2 Example of ModelDescription.xml output variable

By modifying the model description file of the FMU, it can be configured to receive and transmit any number of signals of any type supported by the FMI specification, with exception of enumerations. Hence, it can be reused in other demos and setups. However, changing the configuration for the demo described in this document will break functionality.

Table 1 Control System Communication Inputs and Outputs

Name	Unit/Dimension	Causality
Measured_Position[1]	M	Input
Measured_Position[2]	M	Input
Measured_Position[3]	Rad	Input
Measured_Velocity [1]	m/s	Input
Measured_Velocity [2]	m/s	Input
Measured_Velocity [3]	rad/s	Input
Measured_Wind[1]	m/s	Input
Measured_Wind[2]	m/s	Input
Measured_Wind[3]	rad/s	Input
Commanded_Thrust[1]	N	Output
Commanded_Thrust[2]	N	Output
Commanded_Thrust[3]	Nm	Output

2.3 CURRENTMODEL.FMU

The current model FMU simulates a 2D surface current that is used by the vessel dynamics to estimate drag and resistance. Current can be enabled or disabled by setting a parameter on the FMU.

The FMU is based on the Simulink diagram shown in Figure 3 and its interface is described in Table 2 and Table 3. Notice that the current velocity vector has six entries, however, only the x and y components are relevant. The other components are set to zero. The only tunable is a Boolean for enabling and disabling the current.

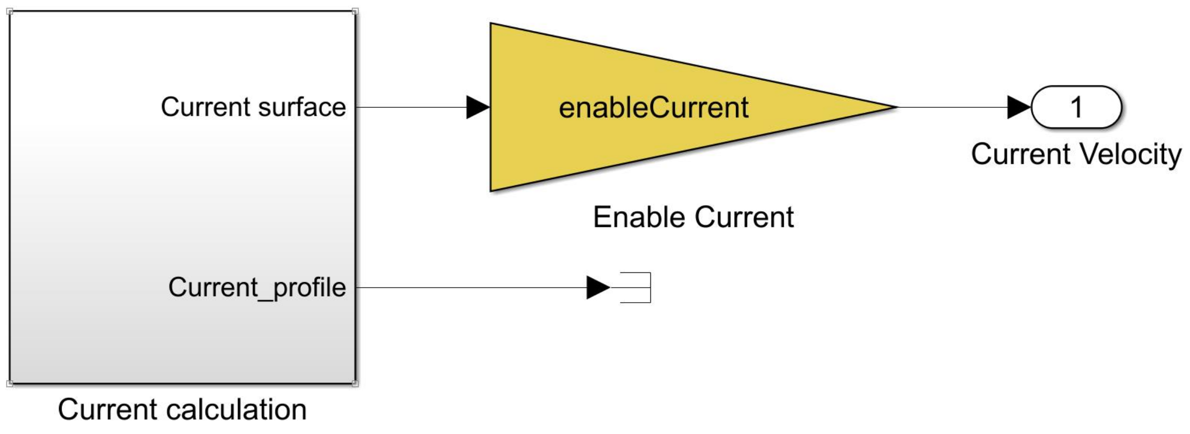

Figure 3 Current Model Interface

Table 2 Current Model Inputs and Outputs

Name	Unit/Dimension	Causality
Current Velocity	[m/s, m/s, m/s, rad/s, rad/s rad/s]	Output

Table 3 Current Model Tunables

Name	Unit/Dimension	Causality
Enable Current	Boolean	Parameter

2.4 DPCONTROLLER.FMU

The DP Controller FMU is an implementation of Gunnerus' DP control system. Internally it is divided into a PD controller, an integrator and a reference feed-forward system. Each part of the control system can be disabled or enabled by changing FMU parameters.

The controller inputs are the vessel's position in NED, velocity in BODY, position setpoint in NED, velocity setpoint in BODY and acceleration setpoint in BODY. The acceleration setpoint can be dropped if the reference feed-forward part of the model is disabled.

A complete list of inputs, outputs and tunables are shown in Table 4 and Table 5.

Table 4 DP Controller Inputs and Outputs

Name	Unit/Dimension	Causality
Vessel Position	[m, m, rad]	Input
Vessel Velocity	[m/s, m/s, rad/s]	Input
Position Setpoint	[m, m, rad]	Input
Velocity Setpoint	[m/s, m/s, rad/s]	Input
Acceleration Setpoint	[m/s^2, m/s^2, rad/s^2]	Input
Commanded Thrust	[N, N, Nm]	Output

Table 5 DP Controller Tunables

Name	Unit/Dimension	Causality
EnablePDControl	Boolean	Parameter
EnableIAction	Boolean	Parameter
EnableRefFF	Boolean	Parameter

2.5 REFERENCEMODEL.FMU

The reference model creates smooth setpoint signals for the DP controller, ensuring a controlled and stable operation. The signals are based on the desired position provided by the box reference FMU.

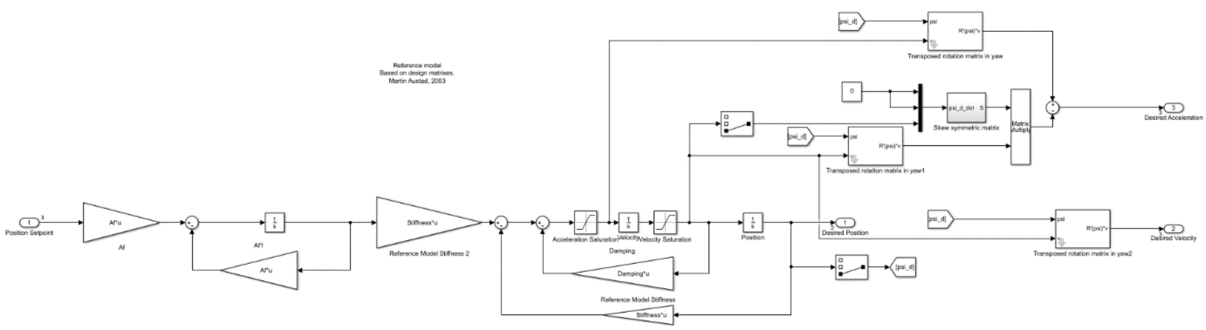

Table 6 Reference Model Inputs and Outputs

Name	Unit/Dimension	Causality
Position Setpoint	[m, m, rad]	Input
Desired Position	[m, m, rad]	Output
Desired Velocity	[m/s, m/s, rad/s]	Output
Desired Acceleration	[m/s^2, m/s^2, rad/s^2]	Output

2.6 SIMULATORCOMMUNICATION.FMU

The SimulatorCommunication.fmu is a communication subsystem that handles communication between the simulator and control system on the control system side.

Table 7 shows the input and output signals of the communication FMU. Table 7 Simulator Communication Inputs and Outputs

Table 7 Simulator Communication Inputs and Outputs

Name	Unit/Dimension	Causality
Commanded_Thrust[1]	N	Input
Commanded_Thrust[2]	N	Input
Commanded_Thrust[3]	N/m	Input
Measured_Position[1]	m	Output
Measured_Position[2]	m	Output
Measured_Position[3]	rad	Output
Measured_Velocity[1]	m/s	Output
Measured_Velocity[2]	m/s	Output
Measured_Velocity[3]	rad/s	Output
Measured_Acceleration[1]	m/s^2	Output
Measured_Acceleration[2]	m/s^2	Output
Measured_Acceleration[3]	rad/s^2	Output
Measured_Wind[1]	m/s	Output
Measured_Wind[2]	m/s	Output
Measured_Wind[3]	rad/s	Output

2.7 ThrusterDynamics.fmu

Thruster dynamics are implemented by the Thruster dynamics FMU. Its interface and implementation are shown in Figure 4.

The model can be configured to simulate thruster dynamics using a simple first order system or to pass the commanded thrust directly to the DP controller. It is also possible to enable thrust saturation.

For a list of outputs, inputs and tunables, see Table 8 and Table 9.

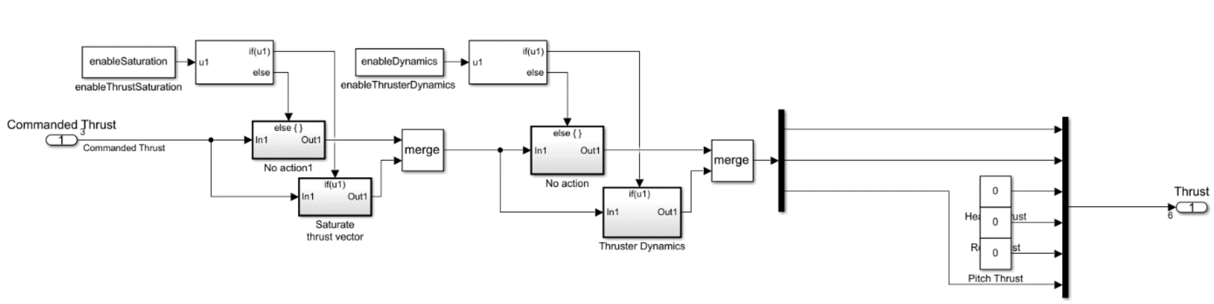

Figure 4 Thruster dynamics subsystem

Table 8 Thruster Dynamics Inputs and Outputs

Name	Unity/Dimension	Causality
Commanded Thrust	[N, N, Nm]	Input
Thrust	[N, N, N, Nm, Nm, Nm]	Output

Table 9 Thruster Dynamics Tunables

Name	Unity/Dimension	Causality
enableThrustSaturation	Boolean	Parameter
enableThrusterDynamics	Boolean	Parameter

2.8 VESSELMODEL.FMU

Vessel dynamics are implemented by the vessel model FMU. Its interface is shown in Figure 5. The sum forces input has been split into wave forces and thrust and current velocity must be provided by an external simulation model. In this case the current model FMU.

Based on these inputs the FMU produces vessel velocities and accelerations in the BODY frame as well as position in the NED frame.

The original model included both Coriolis and fluid memory effects, but they had to be removed to export the model as an FMU. In addition, the only non-linear damping contributions are cross flow drag and surge.

Additional external forces can be connected to either the wave forces or thrust port. Internally, they are both connected to the same summation block. The current velocity port only considers the x and y components of the input vector. The rest of the entries can be set to zero.

The full list of inputs, outputs and tunables are shown in Table 4 Table 5.

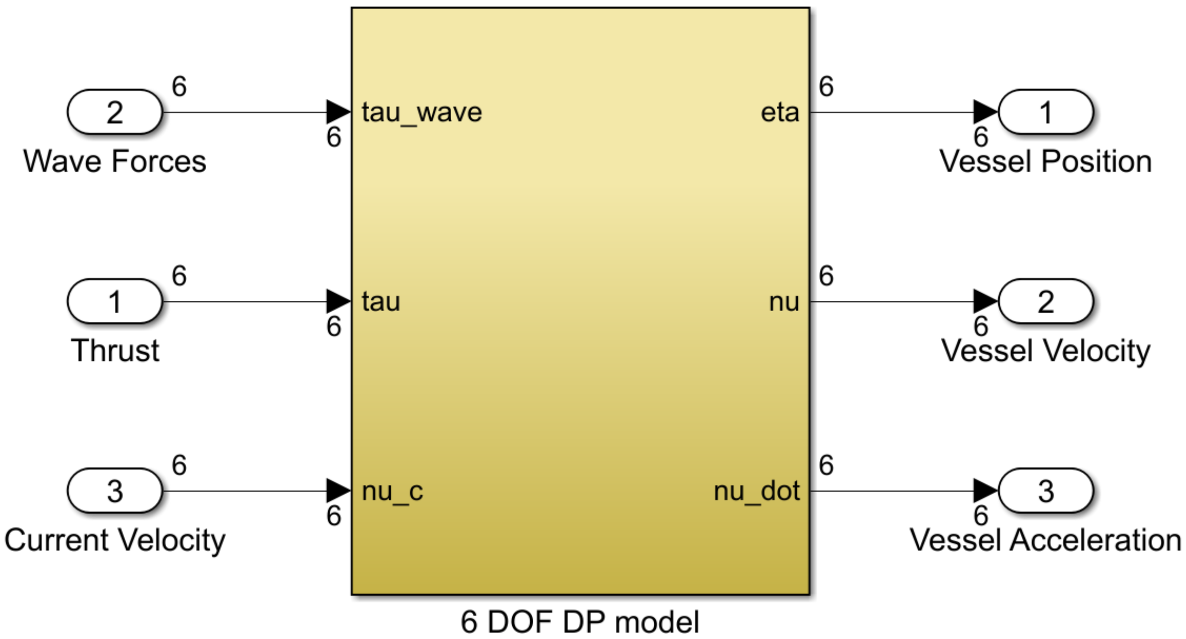

Figure 5 Vessel Model

Name	Unit/Dimension	
Wave Forces	[N, N, N, Nm, Nm, Nm]	Input
Thrust	[N, N, N, Nm, Nm, Nm]	Input

Current Velocity	[m/s, m/s, m/s, rad/s, rad/s, rad/s]	Input
Vessel Position	[m, m, m, rad, rad, rad]	Output
Vessel Velocity	[m/s, m/s, m/s, rad/s, rad/s rad/s]	Output
Vessel Acceleration	[m/s^2, m/s^2, m/s^2, rad/s^2, rad/s^2, rad/s^2]	Output

3 SIMULATOR CONFIGURATIONS

The simulator is distributed as a zip file containing the FMUs, CSE and system description files.

The FMUs are in the fmu folder at the root of the archive.

The system description file for running the centralized simulation is in folder named single-system. The configuration files for the distributed simulations are in the control-system and simulator folders.

The following sections will explain how to run the centralized and distributed simulations and specify hardware requirements.

3.1 RUNNING AS SINGLE-SYSTEM ON ONE COMPUTER

Error! Reference source not found. shows the different FMUs with inputs and outputs connected for single system operation. In single-system operation, both the controller and simulator are running in the same CSE instance thus requiring no communication FMUs.

The simulation is set up for DP operation, where the setpoints are defined in the BoxReference fmu. This FMU defines four position moves of 40 meters which in total defines a box. It is also possible to define setpoints by overriding the Position_Setpoint inputs to the ReferenceModel, either manually or by using the provided Scenario file.

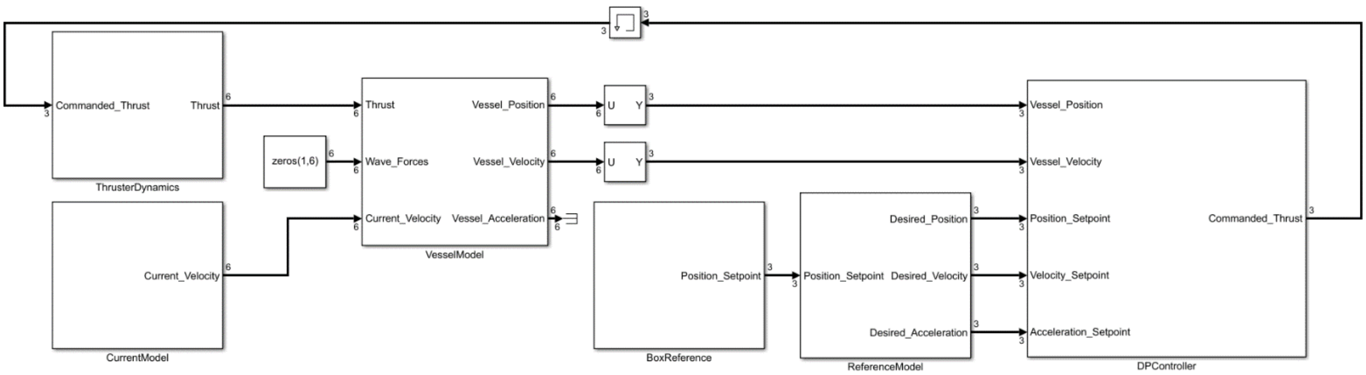

Figure 6 Single system operation

3.1.1 Setup

This section specifies hardware requirements and instructions for running the centralized simulation.

Required hardware

1x Computer

Instructions

1. Start the run-windows.cmd script in the server folder. The web interface should open in your default browser.

2. Point configuration to config/single-system

3. Enable real time factor and press play.

3.2 RUNNING THE CONTROLLER AND SIMULATOR MODULES ON SEPARATE COMPUTERS

The simulation can also run the control system and simulator as separate instances. This requires two computers communicating over a network.

The control system is shown in Figure 7 and is made up by the reference model, DP controller, reference generator and network communication. The vessel simulator is shown in Figure 8 and is made up by the vessel dynamics, current model, thruster dynamics and network communication.

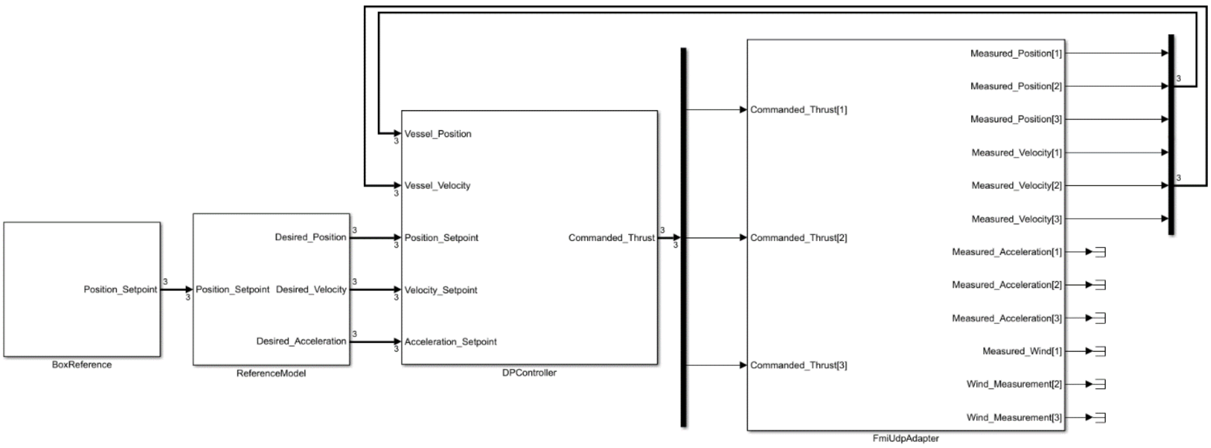

Figure 7 Control system modules

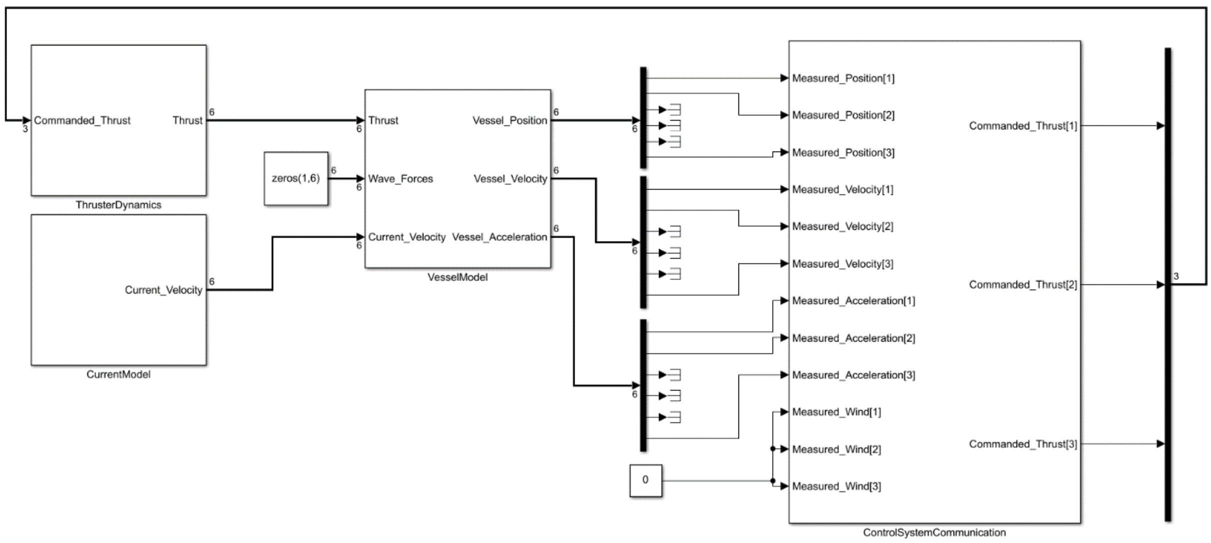

Figure 8 Simulator modules

3.2.1 Setup

Running the distribution simulation requires more hardware and configuration than running the centralized version. Most importantly a second computer and an ethernet cable.

Required hardware

2x computers running Windows

1x ethernet cable

Setup

1. Set up both computers with fixed IPs and connect them using an ethernet cable

2. Open the SimulatorCommunication.fmu and open the file Config.json in the resources folder for editing

3. Change the receive address field to the IP of the computer running the control system and the transmit address to the IP of the simulator computer.

4. Repeat step 2 and 3 for the ControlSystemCommunication fmu.

5. Transfer a copy of the demo files to each computer

6. On the control system computer: run the run-windows.cmd script in the server folder. The web interface should open in your default browser.

7. Point configuration to config/control -system.

8. Enable real time target and press play.

9. On the simulator computer: run the run-windows.cmd script in the cse-server folder. The web interface should open in your default browser.

10. Point configuration to config/simulator

11. Enable real time target and press play.