TMR4240 Project Part 1:
Design of Dynamic Positioning System
Control System Design

Department of Marine Technology
Norwegian University of Science and Technology

August 28, 2023

General Information

The report from the first part of the project must be submitted on Blackboard by October 8th, 2023, including a zip-file containing executable MATLAB/Simulink code/diagrams for all simulations performed by you.

The Project Part 1 will be evaluated (as for the Project Part 2), so the quality of the answer and presentation in the report will be important. Special importance should be given to how the output data is presented.

One single answer will be accepted for each group.

If, for some reasons, some parts cannot be solved, it is important to underline difficulties and thoughts. So, no worries if some difficulties come, the important is to justify the process and underline your attempts to face them, in order to show that you have learned something from it (and tried).

The main objective of the first part in this project is to get you familiar with the MATLAB/Simulink system.

System Overview

This project is developed based on the MSS Toolbox (Marine Systems Simulator). The Marine Systems Simulator (MSS) is a MATLAB/Simulink library and simulator for marine systems. It includes models for ships, underwater vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for real-time simulation. The files for the project can be found on Blackboard and on the MSS Toolbox homepage (see section Getting Started).

Don't be afraid to modify the existing components present on the simulator, since its code is being provided for the TMR4240 course and it is intended for the students to modify anything that is necessary to reach the desired goals.

The provided files will present you a supply ship without the control part, where you will implement it.

Assignment Task

Project Part 1 consists of designing and validating the following systems for a DP positioned SOV:

- Current load
- Controller
- DP Reference model

You are free to use any reasonable methodology to design the mentioned subsystems.

The system overview (as well as the Simulink main diagram) is presented in Figure 1. Note that the blocks that need to be implemented are the gray blocks in the diagram.

Getting Started

It is necessary to install the MSS toolbox before you run the projects file. Download `MSS.zip`, unzip and move to desired destination. Set path to folder and subfolder by right clicking on MSS folder in *Current Folder* box in Matlab.

To initialize the data you need to run the `init.m`, and to run the Simulink model just open the file `part1.slx` and run it. Those two files are the only ones in the simulator folders that you will need to modify.

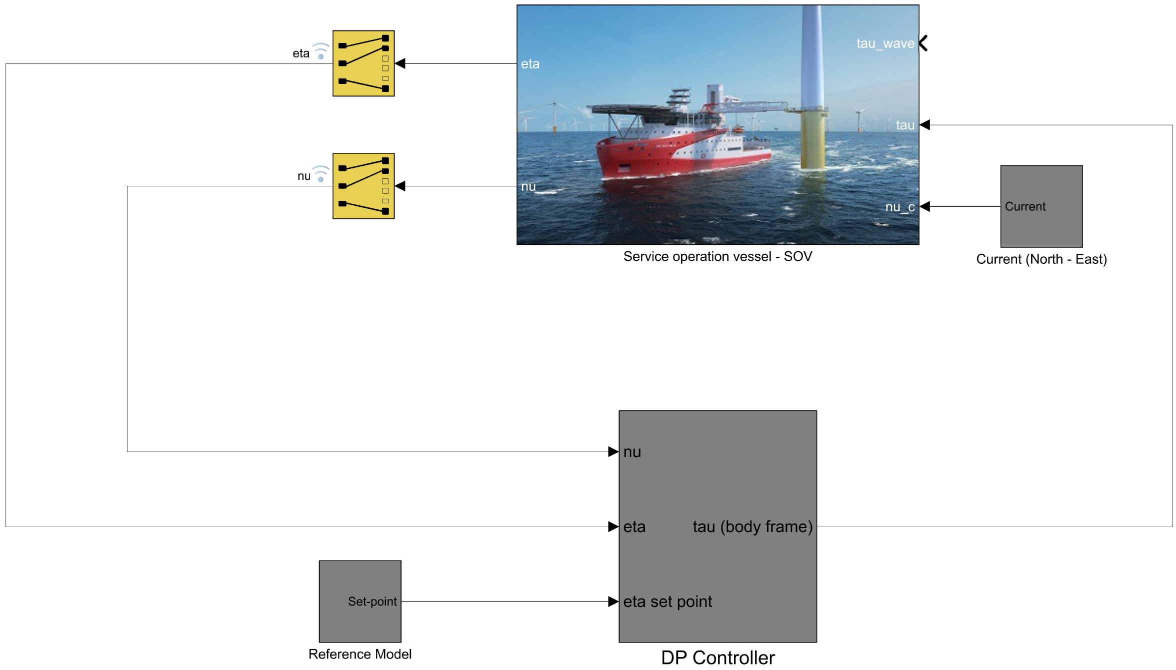

Figure 1: Overview of the system.

1 Current Velocities

The first part is the implementation of the current drag effects on the vessel. The current shall be defined in the NED frame, but only the North and East coordinates are needed (the current average down component will always be 0).

The current block is shown in Figure 1. Note that the actual state of this block just gives a constant zero value in both directions.

The effects of the current forces/moment don't need to be calculated, since the vessel model already accounts for the drag effects.

Hint: Remember that all inputs to the vessel must be in body frame.

2 Controller

The controller shall be responsible for the vessel dynamic positioning. It takes the vessel to the desired set-point and then keep it on the right position, given the environmental loads (which are typically unknown to the DP).

This block shall receive information about the vessel actual position (North, East and heading), set-point (also N, E and Ψ) and calculate the desired forces in surge and sway as well as Ψ moment. All units shall follow the SI (N and Nm).

It is up to you what type of controller will be implemented (linear, nonlinear, hybrid, PID, LQG, etc...).

Present the tuning values put into your controller.

Hint1: Remember that you are working with two different reference frames in this block, so it is important to not mix up NED with body frame.

Hint2: Some examples for controller tuning will be presented in Lecture 5 Dynamic Positioning, Controller Design.

3 Reference Model

It is necessary to generate a reference model for the DP control system, in case that the set-point is changed or the vessel is following a track instead of station keeping.

The initial reference model is just a constant value, it shall be filtered to accept step values as inputs.

Present the tuning values put into your reference model.

4 Mandatory Tests and Report

Several tests are required to show that your system is working properly. They are described in the following sections.

4.1 Simulation 1

For the first simulation, show the DP capabilities of your DP system. With the DP set-point set to [0 0 0] and current set to 0.5 [m/s], going to southeast. Plot the position and heading until you reach steady state, both as individual results and in a xy-plot.

4.2 Simulation 2

Now, make the current vary linearly from 0.5 [m/s] coming from North to 0.5 [m/s] coming from East, while keeping the vessel at the origin [0 0 0]. Plot the position and heading until you reach steady state, both as individual results and in a xy-plot.

4.3 Simulation 3

Compare the different results for a vessel position over time for initial position $\eta_0 = [0\ 0\ 0]$ and $\eta_{\text{SP}} = [10\ 10\ \frac{3\pi}{2}]$, with and without a reference model. Both simulations shall be performed without environmental forces. Plot the position and heading until you reach steady state, both as individual and in a xy-plot. In addition, put the desired trajectory in the individual plots. If the reference model contains velocity trajectories, then plot these with the actual velocities.

4.4 Simulation 4

The last mandatory simulation is to perform a DP 4 corner test. Perform this simulation without environmental forces, and go through the following set-points:

- $\eta_0 = [0\ 0\ 0]$
- $\eta_1 = [50\ 0\ 0]$
- $\eta_2 = [50\ \ -50\ 0]$
- $\eta_3 = [50\ \ -50\ \ -\frac{\pi}{4}]$
- $\eta_4 = [0\ \ -50\ \ -\frac{\pi}{4}]$
- $\eta_5 = [0\ 0\ 0]$

The vessel must keep a steady position on the desired set-point before moving on to the next.

Plot the position and heading until you reach steady state, both as individual and in a xy-plot. In addition, put the desired trajectory in the individual plots. If the reference model contains velocity trajectories, then plot these with the actual velocities.

Tips

- **Expectation:** Max position deviation from setpoint should be 1m - 5m. If the deviation is 10m, there might be some implementation issues in Simulink or bad tuning. If this position deviation is too large, discuss what can be the problem.
- Even though you are in a group of four students, remember that you have other duties as well, so don't try to start solving this project on the last weekend (or last day), since it might be surprisingly time consuming.
- Do not try to solve everything at once. When you develop a big system like this one, work block by block.
- One of the goals for this project is to be similar to what you might find in your professional life. With this in mind, remember that it is not only important to develop a good product, but it is also important to present it properly, with a suitable user interface and well written report.

General Project Report Requirements

The project should be written in a short report. After Project Part 1, the report will be updated with Project Part 2. It means that at the end there is only one report. The following sections should be presented (dependent on which Project part):

- Introduction (to the problem)
- Process plant model
  - Mathematical model of the controlled vessel
- Control plant model
  - Simplified mathematical model of the vessel
- Current model
- Wind model (only part 2)
- Wave model (only part 2)
- Reference model
- Observer design (only in part 2)
  - What type of observer you choose and state the reason for your choice
  - How do you tune your the observer, how do you calculate the observer gain?
- Controller design
  - What type of controller you choose and state the reason for your choice
  - How you tune the controller, how do you calculate the controller gain?
- Simulation results
  - Present different simulation scenarios
  - Scenario 1 $\rightarrow$ n: plot necessary plots
  - Discussion on the simulation results for different scenarios. The most important aspect is whether the chosen controller has fulfill the control objective? If not, state the reason. Have to tried to push the controller/observer/reference gains higher, and what is the consequence of doing that? Etc.
- Conclusion (only in part 2)
  - Summary of the project
    - What have you done in the project?
    - Have you achieved the requirements from the project?

- Provide an overview of
  * The new knowledge or information discovered (Any take-away from the project?)
  * Is there any limitations in your project?
  * Is there anyway to improve the project? (to overcome the limitation)

It is noted that the content of the report from Project Part 1 will only contain the relevant sections and does not need to have the conclusion section. However, Project Part 1 report should contain the discussion(s) to the simulation(s) performed for Part 1. The report after Project Part 2 will contain all sections and the results from simulations in Project Part 2.

Logging Data

There are three common methods to log data, and you are free to chose the one that works best for you, but only two of them are suggested in this project. They are:

- Simulation data inspector
- Export to workspace
- Scope (not recommended)

In this document I will give a brief introduction to the simulation data inspector, since the other methods were already described in Assignment 4.

Simulation Data Inspector

To use this method you need to follow two simple steps:

1. Enable signal logging
2. Define which signals are logged.

These instruction are made on an older version of MATLAB/Simulink, so newer versions of MATLAB/Simulink can skip step 1.

To enable signal logging you have to check the following option:

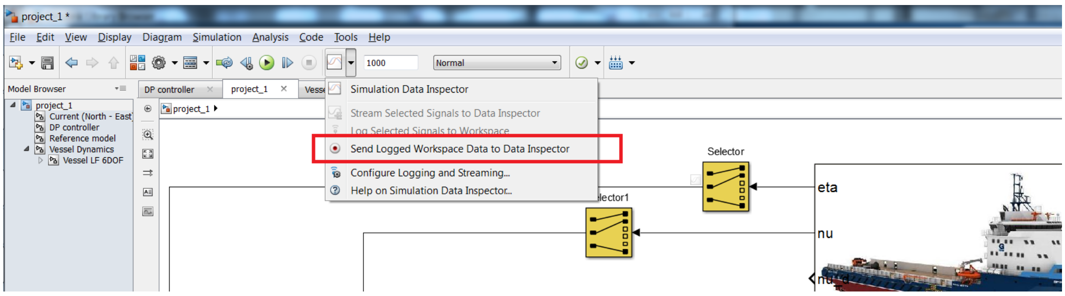

Figure 2: Enabling signal logging, the ”Send Logged Workspace Data to Data Inspector” shall be selected.

After you make sure that the signal logging is enabled (there will be a recording symbol on top of the simulation data inspector symbol), you have to define which signals you will log. Keep in mind that you can log as many signals as you want, but if your system is too big and too many signals are logged, it might take too much time to run and too much space to save the logged data. Right click the line that you want data to be logged and after opening the properties menu you should name the signal and check the option ”log signal data”.

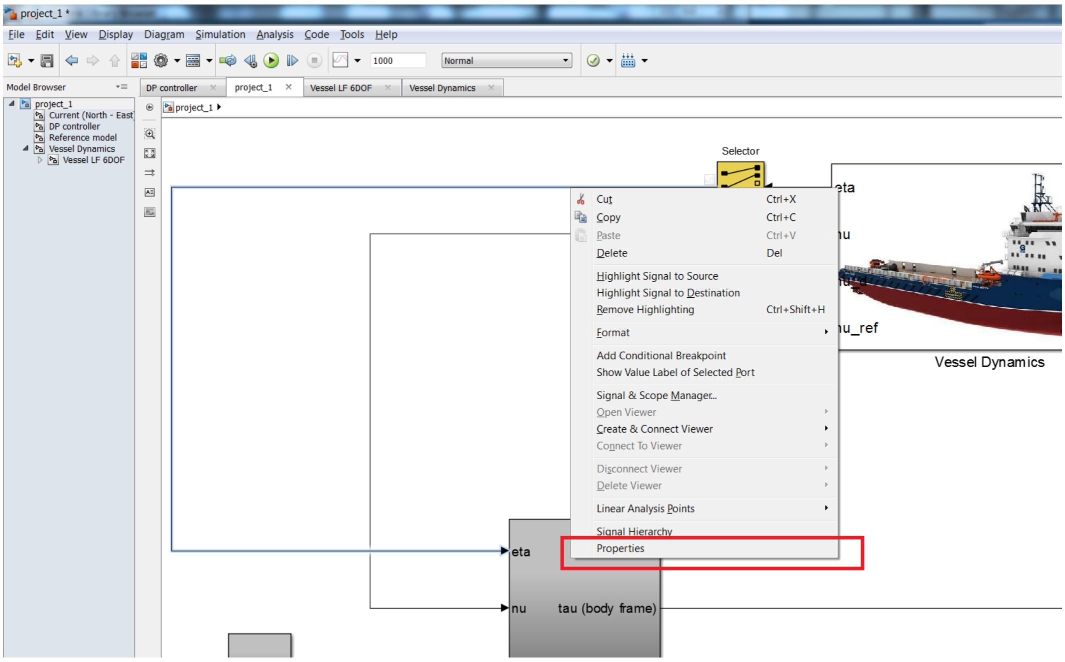

Figure 3: Open the line properties. Note that you will log the data going trough the selected line.

After selecting the signals to be logged, your system will show a small blue antenna on the logged lines, as shown in Figure 5. Remember, you can have as many as those as you need.

Finally, the data can be accessed trough the simulation data inspector (the button on the top with the graph symbol). The output should look like the one in Figure 6. You can compare as many simulations and signals as needed.

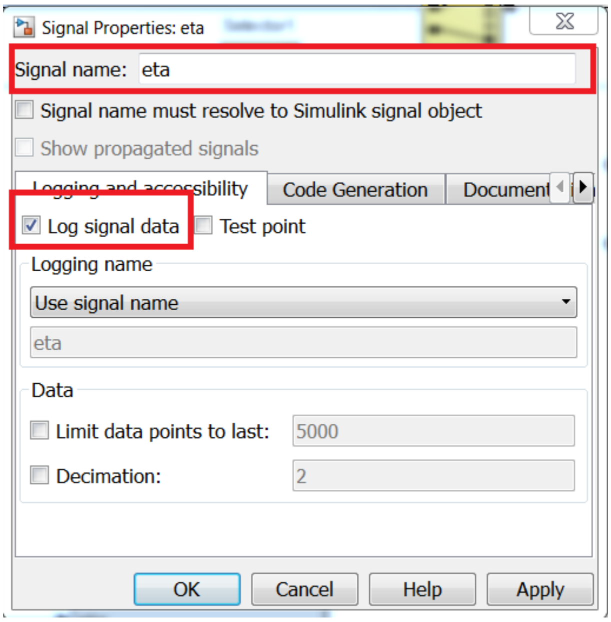

Figure 4: Name the selected signal (with a name that you will remember later) and check the field ”log signal data”.

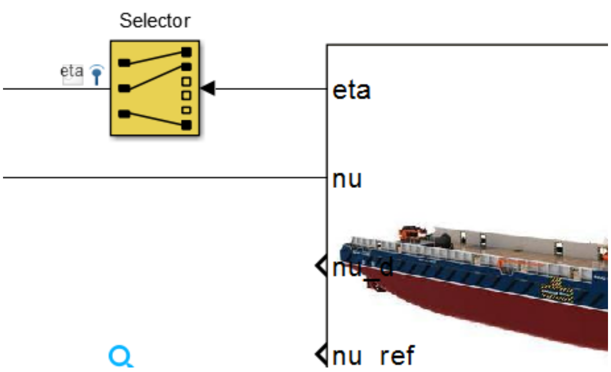

Figure 5: If everything was done correctly, you will see the ”blue antenna” symbol close to the signal name.

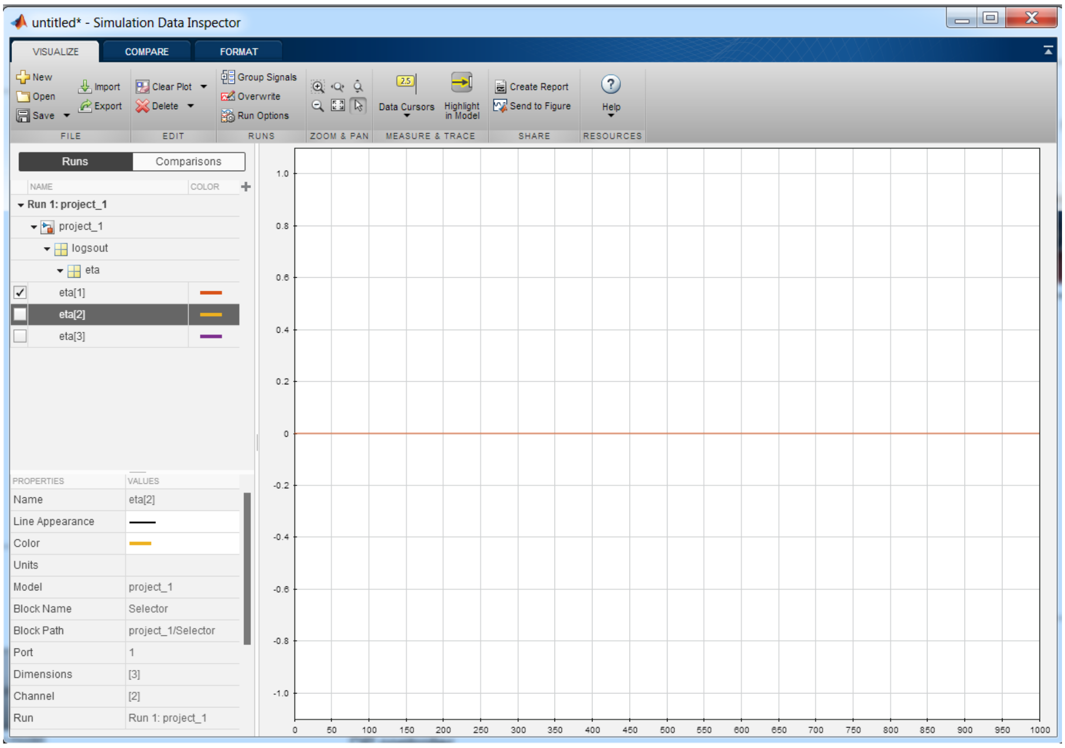

Figure 6: After your simulation is done you can open the simulation data inspector and see all the logged signals, in this example the vessel NORTH position was constant at the origin.