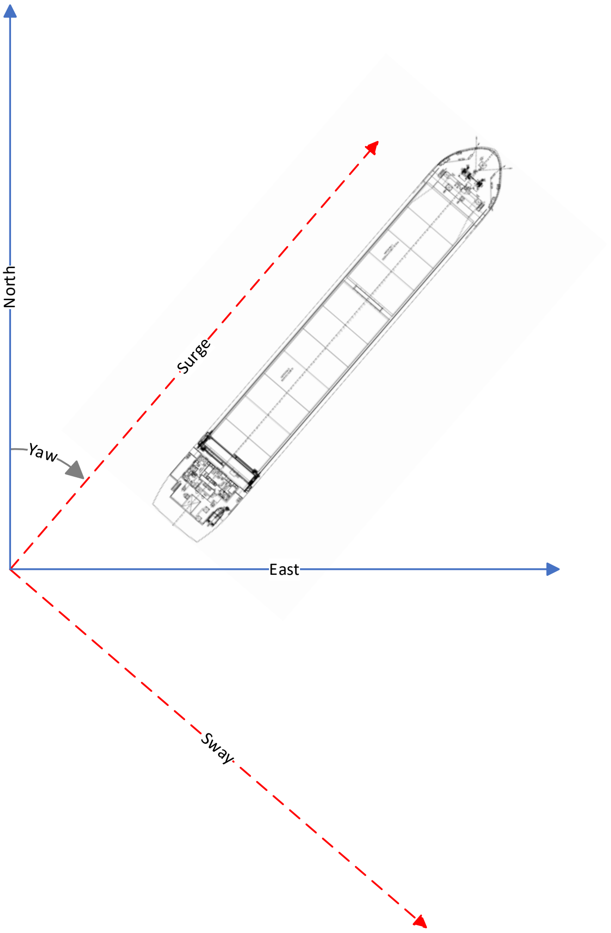
University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2023

Industrial IT and Automation

Dynamic positioning and control of vessels



Kim Hoftvedt

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of
South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2023

Title: Dynamic positioning and control of vessels

Number of pages: 38

Keywords: DP, dynamic positioning, control, MATLAB

Student:	Kim Hoftvedt
Supervisor:	David D. Ruscio
External partner:	None

Summary:

Dynamic positioning systems are an important tool for offshore vessels, particularly rigs and cable or pipe laying ships. Jens Glad Balchen was an important figure in early cybernetics, and he released a paper in 1980 covering mathematical models for simulating the vessel behavior, as well as filter and control algorithms for controlling it. This paper seeks to re implement Balchens theory in MATLAB and simulate and analyze the results. In addition to this, the original model has also been simplified to a linear model.

Preface

I am a student at the University of South-Eastern Norway, and this document is my master thesis delivery. Aside from my education, I have a strong interest for automatic control and programming, and in particular the combination of the two, solving computational problems in elegant and efficient ways.

Porsgrunn, 10.05.23

Kim Hoftvedt

Contents

Preface....................3
Contents....................4
Nomenclature....................5
1 Introduction....................6
1.1 Previous Research....................6
2 Mathematical models....................7
2.1 Low Frequency Model....................7
2.2 High Frequency Model....................8
2.3 Wind Force....................9
2.4 Model Output....................9
2.5 Numerical values....................10
2.6 Coordinate frame translation....................11
2.7 Filter & State observer....................11
2.8 Control....................13
3 Implementation....................15
4 Analyses....................17
5 Simulation results....................18
5.1 Low frequency model....................19
5.2 High Frequency model....................26
5.3 Combined model....................32
5.4 Simplified Model....................33
6 Conclusion....................36
References....................37
Appendices....................38

Nomenclature

DP	Dynamic positioning
NED	North, East, and Down coordinate system
LQ	Linear Quadratic
LF	Low Frequency
HF	High Frequency

1 Introduction

Dynamic positioning systems is a computer system typically designed for ships and offshore rigs that need precise positioning and movement at sea. A DP system is tasked with maintaining a position, heading and/or velocity for a vessel that require precision offshore. These vessels are typically drilling or diving rigs, and cable or pipe laying ships. The challenges surrounding DP systems is primarily the environment combined with measurement imperfection. Wind and water current inflict significant forces on the vessel and thus disturbing the acceleration, velocity and position. One of the earliest work published on dynamic positioning theory was done by Balchen et al (1980) [1]. Jens Glad Balchen was a Norwegian pioneer in cybernetics, with a particular interest in dynamic positioning of offshore vessels.

There are two primary goals of this paper. First is to reconstruct the model, observer and controller presented in Balchens original 1980 paper in modern simulation software. And the second goal is to simulate and evaluate this model, and a simplified form of it.

In order to achieve these goals, Balchens original theory has been thoroughly examined and laid out in detail in chapter 2. Chapter 3 covers how the mathematical models have then been programmed into MATLAB, structured in a way designed for ease of simulating multiple scenarios simultaneously for comparison. An analysis of the model is presented in chapter 4, which makes the basis for the model simplification. The models have then been run through multiple scenarios with the results presented and examined in chapter 5.

1.1 Previous Research

Dynamic position has been a subject of interest for many decades, and there is plenty of published researched on it. There is however not a lot that is directly relevant to the goals of this paper. Or in other words, related to Balchens 1980 paper. But some are worth to mention.

The first paper published on dynamic positioning carrying Balchens name following the one that is the subject of this paper is the 1983 paper titled “Design and Analysis of a Dynamic Positioning System Based on Kalman Filtering and Optimal Control” with Steinar Sælid as main author and co-written by Nils A. Jenssen and Jens G. Balchen [2]. Because this paper is very similar in topic and goals as the 1980 paper, with 3 of the 4 original authors, and released merely 3 years later, this could be viewed as a direct follow-up. The structure presented in this paper has strong similarities with the original, where both is based around having two models for describing the vessel, one for low frequency and another for high frequency. But the models themself has been significantly changed. The “new” low frequency model is a modified version of the original designed to be easier to analyze and “remove a tendency toward very low-frequency oscillation”. There is also a new high frequency model that is said to improve on the original by “introducing a damping term in the oscillators and employing a more robust parameter estimation algorithm”.

In 2022 master student Nour Mohamad Bargouth at the University of South-Eastern Norway did a thesis on dynamic positioning and control [3]. Bargouths thesis is primarily focused around evaluating different control theory but applies it to a simplified version of Balchens original low frequency model. Bargouths does a thorough stability analysis of the low frequency model, as well as contribute with some missing pieces from the original work.

2 Mathematical models

Balchens vessel model consists of primarily two state models, one for low frequency motion and disturbance and one for high frequency disturbance. The low frequency includes current, thrust and wind force effects. The high frequency model emulates rapid changing wind and waves. Both describe motion in surge, sway and yaw, in the vessel coordinate frame.

The model is expressed in two different coordinate frames. The world coordinate frame is static with origo in some arbitrary static point, with axis in north and east orientation. The other coordinate frame is the vessel coordinate frame. The vessel coordinate frame has origo located in the same location as the world coordinate frame, but the vessel coordinate frame is rotated to align with the vessel, with axis in sway and surge directions. Wind and current measurements, as well as reference points is given in world coordinate frame. The mathematical model is expressed in the vessel coordinate frame. For synergizing the different coordinate frames, the data needs to be translated with a translational matrix. Figure 1 shows how the two coordinate frames could look like.

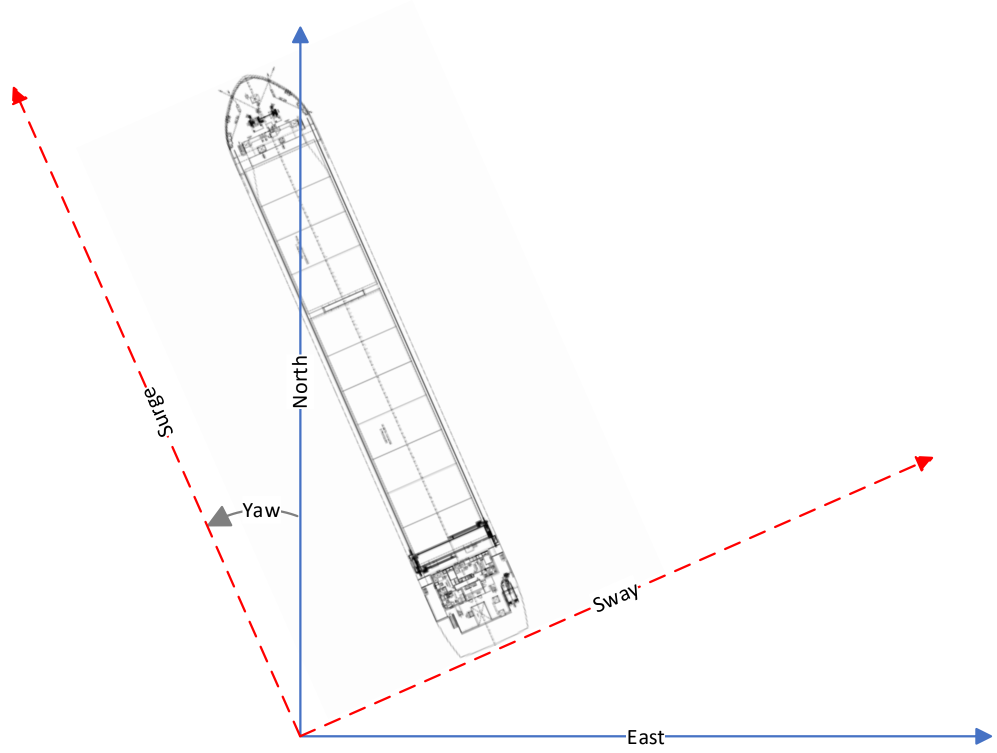


Figure 1 NED & ship frame illustration

2.1 Low Frequency Model

The low frequency model takes thrust, wind force and current speed as inputs, and models the primary component to the vessel speed and position. The low frequency model is given below in equations (2.1) to (2.6).

$$\dot{x}_{L1} = x_{L2} \tag{2.1}$$

$$\dot{x}_{L2} = -\frac{d_1}{m_1}|x_{L2} - x_{C1}|(x_{L2} - x_{C1}) + \frac{1}{m_1}(u_1 + v_1) + \eta_{L1} \tag{2.2}$$

$$\dot{x}_{L3} = x_{L4} \tag{2.3}$$

$$\dot{x}_{L4} = -\frac{d_2}{m_2}|x_{L4} - x_{C2}|(x_{L4} - x_{C2}) + \frac{1}{m_2}(u_2 + v_2) + \eta_{L2} \tag{2.4}$$

$$\dot{x}_{L5} = x_{L6} \tag{2.5}$$

$$\dot{x}_{L6} = -\frac{d_3}{m_3}|x_{L6}|x_{L6} - \frac{d_4}{m_3}|x_{L4} - x_{C2}|(x_{L4} - x_{C2}) + \frac{1}{m_3}(u_3 + v_3 + x_{C3}) + \eta_{L3} \tag{2.6}$$

Where x_{ci} is water current speed, u_i is thrust force, v_i is wind force and η_{Li} is zero-mean gaussian white noise. m_i and d_i is vessel parameters, where m_i is inertia coefficients and d_i is drag and momentum coefficients. The states represent the vessel position and velocity in the following order:

x_{L1} – Position in sway

x_{L2} – Velocity in sway

x_{L3} – Position in surge

x_{L4} – Velocity in surge

x_{L5} – Heading (yaw)

x_{L6} – Heading rate

2.2 High Frequency Model

The high frequency model is shown in equation (2.7) to (2.15) and the states represent separate position and velocity to the low frequency model. This is primarily a disturbance model designed to emulate waves and rapidly changing winds.

$$\dot{x}_{H1} = x_{H2} \tag{2.7}$$

$$\dot{x}_{H2} = -\omega_1^2 x_{H1} + \eta_{H1} \tag{2.8}$$

$$\dot{x}_{H3} = x_{H4} \tag{2.9}$$

$$\dot{x}_{H4} = -\omega_2^2 x_{H3} + \eta_{H2} \tag{2.10}$$

$$\dot{x}_{H5} = x_{H6} \tag{2.11}$$

$$\dot{x}_{H6} = -\omega_3^2 x_{H5} + \eta_{H3} \tag{2.12}$$

$$\dot{\omega}_1 = \eta_{H4} \tag{2.13}$$

$$\dot{\omega}_2 = \eta_{H5} \tag{2.14}$$

$$\dot{\omega}_3 = \eta_{H6} \tag{2.15}$$

The high frequency model does not have any feedback from outside sources. This means that the states produced by this model is completely independent on what happens elsewhere. The model primarily consists of 3 sets of a sinusoid in a state space representation, where ω is the frequency [4]. And with the definition of $\dot{\omega}$ equal to random noise, the frequency ω becomes a "random walk". This means that, at least in theory, the frequency of the high frequency model can become any real number with enough time.

x_{H1} – Position in sway

x_{H2} – Velocity in sway

x_{H3} – Position in surge

x_{H4} – Velocity in surge

x_{H5} – Heading (yaw)

x_{H6} – Heading rate

ω_1 – Frequency in sway

ω_2 – Frequency in surge

ω_3 – Frequency in yaw

η_{Hi} – Zero mean gaussian white noise

2.3 Wind Force

Wind force is not described in detail in Balchens paper, only mentioned as a function of the wind speed vector without further elaborations. Because of this the wind force calculations used in this paper is taken from Bargouths's earlier work [3], and given as the equation (2.16), (2.17) & (2.18).

$$v_1 = \frac{\rho V_1^2}{2} \cdot C_x \cos(\varphi) A_f \tag{2.16}$$

$$v_2 = \frac{\rho V_2^2}{2} \cdot C_y \sin(\varphi) A_l \tag{2.17}$$

$$v_3 = \frac{\rho V_3^2}{2} \cdot C_N \sin(\varphi) A_l L \tag{2.18}$$

Here V_i is the relative wind speed component, ρ is the density of the air. C_x, C_y & C_N are wind coefficients. φ is the relative angle between wind direction and vessel heading. A_f is area of vessel front, A_l is area of vessel from the (long) side and L is the length of the vessel. v_i is the wind force and is using the same notation as in the low frequency model.

2.4 Model Output

The output of the model is the two position values and the rotation value, as the sum of the low and high frequency model, with some random error. The output definition is shown in equation (2.19), (2.20) and (2.21). These are converted to world coordinate frame and is everything regarding the vessel that is assumed to be a measurable value for the observer.

$$y_1 = x_{L1} + x_{H1} + \eta_{Y1} \tag{2.19}$$

$$y_2 = x_{L3} + x_{H3} + \eta_{Y2} \tag{2.20}$$

$$y_3 = x_{L5} + x_{H5} + \eta_{Y3} \tag{2.21}$$

Here y_1 is position in north direction, y_2 is position in east direction, and y_3 is rotation around the yaw axis.

2.5 Numerical values

The model presented in Balchens 1980 paper was adapted for a floating off-shore rig named “Seaway Swan” shown in Figure 2, which was at the time used as a diving rig [5]. The model parameters presented in the paper is based on this vessel.

$$m_1 = 2.4 \cdot 10^7 kg$$

$$m_2 = 4.0 \cdot 10^7 kg$$

$$m_3 = 4.7 \cdot 10^{10} kg/m^2$$

The drag coefficients vary as functions of the vessel heading relative to current direction, and they are only presented as graphs without exact values. For the sake of further analyses in this paper these are assumed to be constants roughly around the mean value.

$$d_1 = 5.0 \cdot 10^{-5} kg/m$$

$$d_2 = 21.0 \cdot 10^{-5} kg/m$$

$$d_3 = 1.1 \cdot 10^{-10} kg \cdot m^2$$

$$d_4 = 201.0 \cdot 10^{-15} kg$$

The wind coefficients, area and length of vessel used to calculate wind force is not mentioned in Balchens paper, and might be lost to time as the original vessel was decommissioned decades ago. Thus these values are assumed as the same values used by Bargouths, even though they clearly represent a different vessel.

$$C_x = 0.6$$

$$C_y = 0.8$$

$$C_N = 0.1$$

$$A_f = 500\ m^2$$

$$A_l = 1100\ m^2$$

$$L = 73.2\, m$$

$$\rho = 1.23\ kg/m^3$$

Figure 2 Seaway Swan

2.6 Coordinate frame translation

Because some of the model equations are given in ship frame, while others are in NED frame, a translation matrix is used for converting sets of values from NED frame to ship frame, or from ship frame to NED frame. Equation (2.22) shows the rotational matrix R used to convert a set of 3 states from ship frame to NED frame, and R^T is used in the opposite direction.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

This matrix is used to convert the y array of output states, u vector for thrust, as well as wind and water current vectors. The same matrix can be used in combination with the 6 set of states of the low or high frequency model, but the states must be rearranged into two sets of three states.

2.7 Filter & State observer

Wind speed vector and the position and rotation outputs of the model is assumed to be measurable, but the outputs are considered very noisy and the control schema laid out requires both the full set of low frequency states and current. Thus a Kalman filter is implemented to filter the measured values and estimate the full set of states. Equations (2.23) to (2.28) shows the low frequency state estimator, and the current estimator is shown in equations (2.29),(2.30) & (2.31).

$$\dot{\hat{x}}_{L1} = \hat{x}_{L2} + k_{su1}(y_1 - \hat{x}_{L1} - \hat{x}_{H1}) \tag{2.23}$$

$$\dot{\hat{x}}_{L2} = -\frac{d_1}{m_1}|\hat{x}_{L2} - \hat{x}_{C1}|(\hat{x}_{L2} - \hat{x}_{C1}) + \frac{1}{m_1}(u_1 + \hat{v}_1) + k_{su2}(y_1 - \hat{x}_{L1} - \hat{x}_{H1}) \tag{2.24}$$

$$\dot{\hat{x}}_{L3} = \hat{x}_{L4} + k_{sw1}(y_2 - \hat{x}_{L3} - \hat{x}_{H3}) \tag{2.25}$$

$$\dot{\hat{x}}_{L4} = -\frac{d_2}{m_2}|\hat{x}_{L4} - \hat{x}_{C2}|(\hat{x}_{L4} - \hat{x}_{C2}) + \frac{1}{m_2}(u_2 + \hat{v}_2) + k_{su2}(y_2 - \hat{x}_{L3} - \hat{x}_{H3}) \tag{2.26}$$

$$\dot{\hat{x}}_{L5} = \hat{x}_{L6} + k_{ya1}(y_3 - \hat{x}_{L5} - \hat{x}_{H5}) \tag{2.27}$$

$$\dot{\hat{x}}_{L6} = -\frac{d_3}{m_3}|\hat{x}_{L6}|\hat{x}_{L6} - \frac{d_4}{m_3}|\hat{x}_{L4} - \hat{x}_{C2}|(\hat{x}_{L4} - \hat{x}_{C2}) + \frac{1}{m_3}(u_3 + \hat{v}_3 + \hat{x}_{C3}) + k_{ya2}(y_3 - \hat{x}_{L5} - \hat{x}_{H5}) \tag{2.28}$$

$$\dot{\hat{x}}_{C1} = k_{C1}(y_1 - \hat{x}_{L1} - \hat{x}_{H1}) \tag{2.29}$$

$$\dot{\hat{x}}_{C2} = k_{C2}(y_2 - \hat{x}_{L3} - \hat{x}_{H3}) \tag{2.30}$$

$$\dot{\hat{x}}_{C3} = k_{C3}(y_3 - \hat{x}_{L5} - \hat{x}_{H5}) \tag{2.31}$$

Both the low and high frequency observers share a lot of the exact same math as the model itself, with random noise replaced by estimation error correction. But the calculation for the frequency ω in the HF model holds the biggest difference, as it's calculated from the steady state solution as shown in equation (2.32) & (2.33). These equations show how to calculate the estimate for the first omega (in surge direction), but the same formulation is used to calculate the other two, with states and gains for their respective directions.

$$r_1 = \frac{2\hat{\omega}_1}{(k_{su3} \cdot \hat{\omega}_1)^2 + k_{su4}^2}(k_{su3}\hat{x}_{H2} - k_{su4}\hat{x}_{H1}) \tag{2.32}$$

$$\hat{\omega}_{1,k+1} = \hat{\omega}_{1,k} - r_1 \tag{2.33}$$

The high frequency observer is shown in equations (2.34) to (2.39), and while the estimated high frequency states is not used for control, it is included in the estimation error. It will therefor have an indirect effect on the low frequency estimate. It's also worth noting that Balchen writes that the high frequency estimate is to not be updated until the current and low frequency estimate converges, though without elaborating on any specific criteria.

$$\dot{\hat{x}}_{H1} = \hat{x}_{H2} + k_{su3}(y_1 - \hat{x}_{L1} - \hat{x}_{H1}) \tag{2.34}$$

$$\dot{\hat{x}}_{H2} = -\hat{\omega}_1^2\hat{x}_{H1} + k_{su4}(y_1 - \hat{x}_{L1} - \hat{x}_{H1}) \tag{2.35}$$

$$\dot{\hat{x}}_{H3} = \hat{x}_{H4} + k_{sw3}(y_2 - \hat{x}_{L3} - \hat{x}_{H3}) \tag{2.36}$$

$$\dot{\hat{x}}_{H4} = -\hat{\omega}_2^2\hat{x}_{H3} + k_{sw4}(y_2 - \hat{x}_{L3} - \hat{x}_{H3}) \tag{2.37}$$

$$\dot{\hat{x}}_{H5} = \hat{x}_{H6} + k_{ya3}(y_3 - \hat{x}_{L5} - \hat{x}_{H5}) \tag{2.38}$$

$$\dot{\hat{x}}_{H6} = -\hat{\omega}_3^2 \hat{x}_{H5} + k_{ya4}(y_3 - \hat{x}_{L5} - \hat{x}_{H5}) \tag{2.39}$$

The numerical values given for the Kalman gains K_{su}, K_{sw}, K_{ya} and K_c is shown below:

$$K_{su} = K_{sw} = [0.045, \quad 0.001, \quad 0.3, \quad 0.07] \tag{2.40}$$

$$K_{ya} = [0.145, \quad 0.004, \quad 0.7, \quad 0.15] \tag{2.41}$$

$$K_c = [0.001, \quad 0.001, \quad 2.0] \tag{2.42}$$

2.8 Control

The control Balchen put forth in his paper is an LQG controller using all 6 vessel states. The control does not have integral action, But it does have feed forward on wind forces and water current. It’s designed to minimize the cost function in equation (2.43). Here Δx is the error between reference and (estimated) state for the full set of states, and u is the controller output, thrust in the 3 directions of motion.

$$J = \lim_{T\to\infty} \frac{1}{T} \int_0^T (\Delta x^T q \Delta x + u^T p u)\, dt \tag{2.43}$$

The full thrust calculation is given in equations (2.44), (2.45) & (2.46).

$$u_1 = G_{su} \cdot \begin{bmatrix} \Delta x_{L1} \\ \hat{x}_{L2} \end{bmatrix} + u_{C1} + u_{W1} \tag{2.44}$$

$$u_2 = G_{sw} \cdot \begin{bmatrix} \Delta x_{L3} \\ \hat{x}_{L4} \end{bmatrix} + u_{C2} + u_{W2} \tag{2.45}$$

$$u_3 = G_{ya} \cdot \begin{bmatrix} \Delta x_{L5} \\ \hat{x}_{L6} \end{bmatrix} + u_{C3} + u_{W3} \tag{2.46}$$

In these equations $\Delta x_{L1} = \hat{x}_{L1} - x_{L1}^{ref}$ and equally for Δx_{L3} and Δx_{L5}. It’s worth noting that the same definition is used for the velocity states but in the original paper it is assumed that velocity setpoint is equal to zero, thus raising a question whether these formulas could be altered to have deltas for all the states in combinations with a variable velocity set point.

$$G_{su} = \left[-\sqrt{q/p}, \quad -\sqrt{2m_1\sqrt{q/p}} \right] \tag{2.47}$$

The controller gain G is defined in equation (2.47) but the exact values given is:

$$G_{su} = [-3.0 \cdot 10^4\ N/m, \quad -1.15 \cdot 10^6\ Ns/m] \tag{2.48}$$

$$G_{sw} = [-3.0 \cdot 10^4\ N/m, \quad -1.55 \cdot 10^6\ Ns/m] \tag{2.49}$$

$$G_{ya} = [-2.0 \cdot 10^8\ N/m, \quad -5.0 \cdot 10^9\ Ns/m] \tag{2.50}$$

The wind part of the thrust calculation is defined as $u_{Wi} = -\hat{v}_i$ where $\hat{v}_i$ is the estimated wind force affecting the vessel. The current part u_{Ci} is defined in the equations (2.51), (2.52) & (2.53).

$$u_{C1} = d_1 \cdot |\hat{x}_{L2} - \hat{x}_{C1}| \cdot (\hat{x}_{L2} - \hat{x}_{C1}) \tag{2.51}$$

$$u_{C2} = d_2 \cdot |\hat{x}_{L4} - \hat{x}_{C2}| \cdot (\hat{x}_{L4} - \hat{x}_{C2}) \tag{2.52}$$

$$u_{C3} = d_4 \cdot |\hat{x}_{L4} - \hat{x}_{C2}| \cdot (\hat{x}_{L4} - \hat{x}_{C2}) - \hat{x}_{C3} \tag{2.53}$$

3 Implementation

MATLAB was chosen as the target framework for implementing the models. In addition, it was constructed in class structure in order to make the program more organized as well as more adaptive and flexible during runtime. Building the simulation in a class structure allows much easier use of multiple models with different parameters to be run simultaneously without duplicating code. Figure 3 shows the class diagram and shows their inheritance and main functions.

The Main program is a script that generates all the objects and call their primary functions, and it keeps track of setpoints, states and thrust forces. Everything in the main program is given in world coordinate frame.

The Environment class generates wind and current speeds in meters pr secund and direction. Because wind forces is relative to vessel parameters, the environment class does not compute any forces.

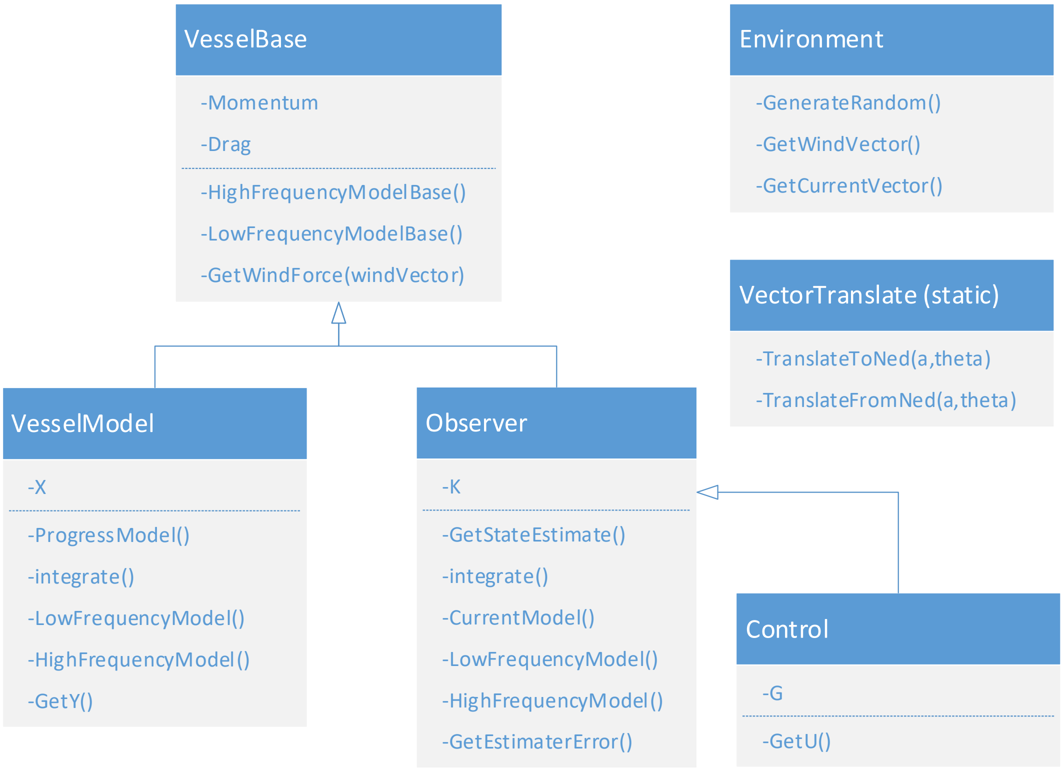


Figure 3 Class Diagram

The “VesselModel” class contains the mathematical models used to describe the vessel movement. It takes parameters and initial states as input on the constructor. “ProgressModel” is the primary method of this class that iterate the model, with delta time, thrust force and environment vectors as input parameters. Internally within the class these inputs are transformed to the ship coordinate frame and the model equations are integrated. The output states are transformed back to world coordinate frame and generate a position array (outputs).

The Controller class needs the full set of estimated states, together with reference value as input to generates desired thruster values, but the method takes the output model vector as input and redirect the array to its base observer class for conversion.

The Observer class takes the model output and estimates the full set of states. Balchens state estimator is a modified version of the model and thus also needs wind speed vector as input, which is assumed to be measurable. And the class also needs and does the rotation of the parameters since the observer calculation is based on the ship coordinate frame.

The “VesselBase” class is the parent class for both the Vessel Model class and the Observer class, and it holds common elements for the two. The “GetWindForce” method is a method converting wind speed vector too wind force vector. The low and high frequency model equations in this class is the common elements from the observer equations and model equations. As these are mostly the same equations with the model adding random noise and the observer adding estimation error gain.

Vector Translate is a class with only static functions to generate rotational matrix and do the transformation. The reason for separating this out into a separate class is to reduce code duplication as several of the other classes needs this functionality.

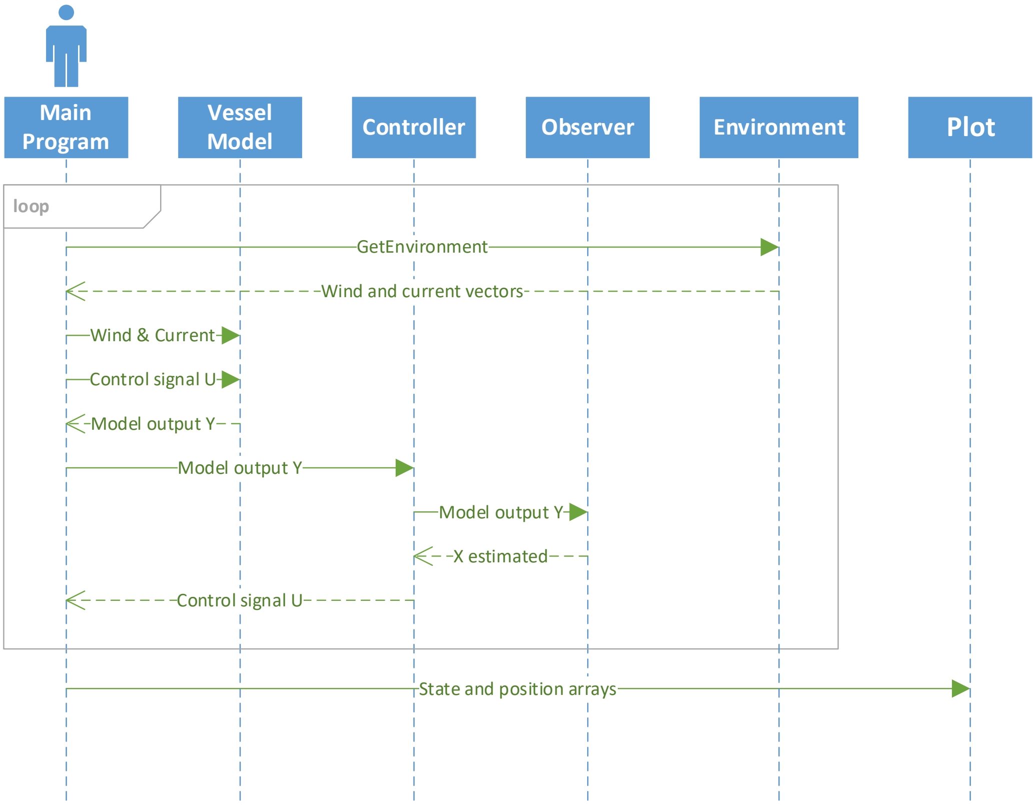


Figure 4 Interaction diagram

Interaction diagram is shown in Figure 4 and illustrates how the main program script interacts with the classes, and how the information flows throughout the program. This is a slightly simplified illustration as there are more method calls than in the diagram, but the purpose of this illustration is to show an overview.

4 Analyses

One of the primary questions raised in this paper is whether the original model presented can be simplified. The root of this question arised from the observation that in the model equations (2.2), (2.4) and (2.6). The first element is a product of d_n/m_k, and because the inertia and drag coefficients covered in chapter 2.5 has a large disparity in their size, the resulting fraction becomes extremely small, and have the following values:

$$\frac{d_1}{m_1} = \frac{5.0 \cdot 10^{-5}}{2.4 \cdot 10^{7}} = 2.08 \cdot 10^{-12}$$

$$\frac{d_2}{m_2} = \frac{21.0 \cdot 10^{-5}}{4.0 \cdot 10^{7}} = 5.25 \cdot 10^{-12}$$

$$\frac{d_3}{m_3} = \frac{1.1 \cdot 10^{-10}}{4.7 \cdot 10^{10}} = 2.24 \cdot 10^{-21}$$

$$\frac{d_4}{m_3} = \frac{201.0 \cdot 10^{-15}}{4.7 \cdot 10^{10}} = 4.28 \cdot 10^{-24}$$

And thus the question on whether or not it makes any difference to the model, and if the model could be used in a simplified form, where all drag coefficients are set to zero, effectively eliminating a chunk of the low frequency model equations. If the drag coefficients where set to zero, it would reduce the low frequency model to the following set of functions:

$$\dot{x}_{L1} = x_{L2} \tag{4.1}$$

$$\dot{x}_{L2} = \frac{1}{m_1}(u_1 + v_1) \tag{4.2}$$

$$\dot{x}_{L3} = x_{L4} \tag{4.3}$$

$$\dot{x}_{L4} = \frac{1}{m_2}(u_2 + v_2) \tag{4.4}$$

$$\dot{x}_{L5} = x_{L6} \tag{4.5}$$

$$\dot{x}_{L6} = \frac{1}{m_3}(u_3 + v_3 + x_{C3}) \tag{4.6}$$

The viability of this simplified model will be explored further in chapter 5.4.

5 Simulation results

Chapter 2 covers everything presented in Balchens paper, but there are some areas left ambiguous. Some of the things that is left unclear by the author, and will affect the results of the simulation is method of integrating the model equations, the time step used, but most notably the standard deviation of the numerous instances of random noise. The random noise is labeled as zero mean gaussian white noise, but the standard deviation of the noise is not mentioned for most of the instances, and different values will affect the outcome of the simulation.

In order to easily test with different standard deviations, and for reference in this paper, an array of 4 elements representing standard deviation in groups of model equations has been implemented and denoted as sigma (σ).

- σ_1 is the standard deviation in the noise in measurement output, η_{Yi} from equations (2.19) to (2.21). This one is mentioned in the paper to be between 0.5 m and 0.7 m.
- σ_2 is the standard deviation in the low frequency model, η_{Li} in equations (2.2),(2.4) & (2.6).
- σ_3 is the standard deviation in the high frequency model, η_{H1}, η_{H2} and η_{H3} in equations (2.8), (2.10) & (2.12).
- σ_4 is the standard deviation in the acceleration of ω_i, η_{H4}, η_{H5} and η_{H6} from equations (2.13) to (2.15).

Although the timestep of the simulation is not described, Balchen writes that the observer and controller is made with the expectation of running with a time step of 1 second, this is assumed to be true for the model simulation as well.

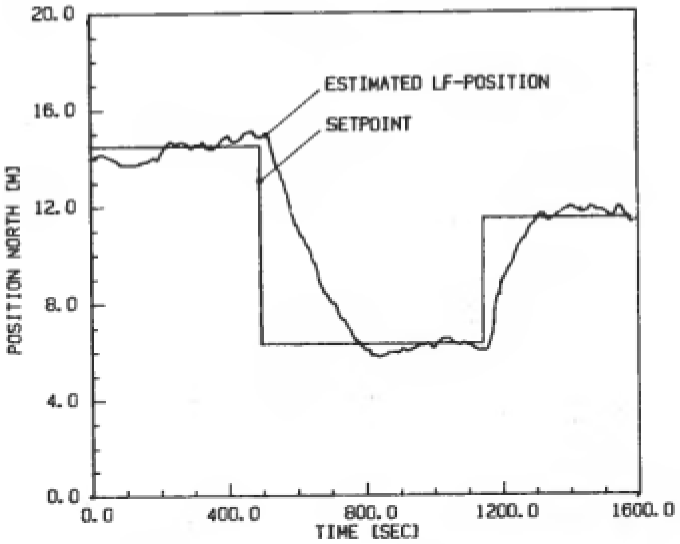


Figure 5 Position in north from Seaway Swan control test

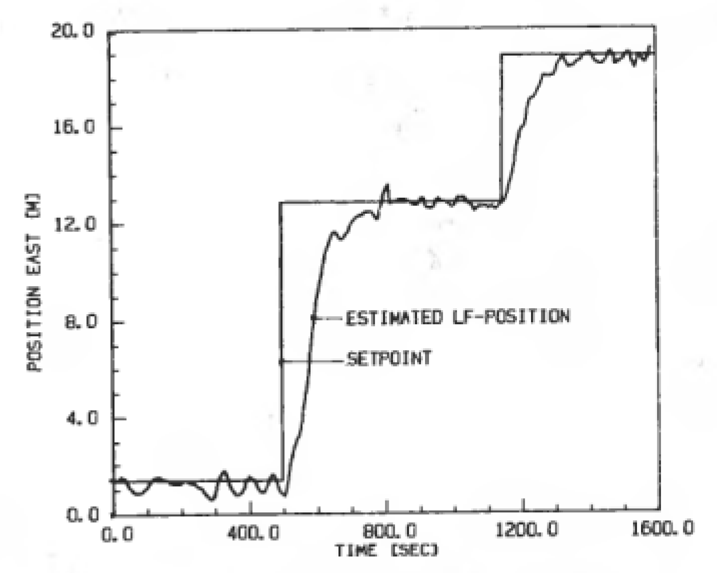


Figure 6 Position in east from Seaway Swan control test

Balchen provided several plots of both simulations and actual control test of the Seaway Eagle & Seaway Swan. Some of these plots will be used as references here to evaluate and compare against this implementation. Figure 5 and Figure 6 show the results provided of the real sea trials of Seaway Eagle, and while this implementation doesn't have the luxury of testing on any real vessel, it gives a baseline for what to expect out of the simulation and control. And to ensure a good comparison the set points and time scale from these plots will be used throughout this chapter.

For the sake of repeatability and comparison between different experiments, the environmental factors are kept static with a wind speed at a reasonable $5\,m/s$ at $45°$ heading in the world coordinate frame, and current at $0.3\,m/s$ at $120°$ heading.

5.1 Low frequency model

The first simulated model is with all standard deviations equal to zero. This means in practice it's only the low frequency model, as the high frequency model is initialized as zero and without any noise influencing it remain at zero. But this is also without noise in the low frequency and output model as well. As seen in Figure 7, Figure 8 & Figure 9 the controller manages the process in a reasonably good way. It's reacting set point fast with some minor overshooting. When the controller tunes in to the set point it is stable, accurate and compensates for water current and wind forces.

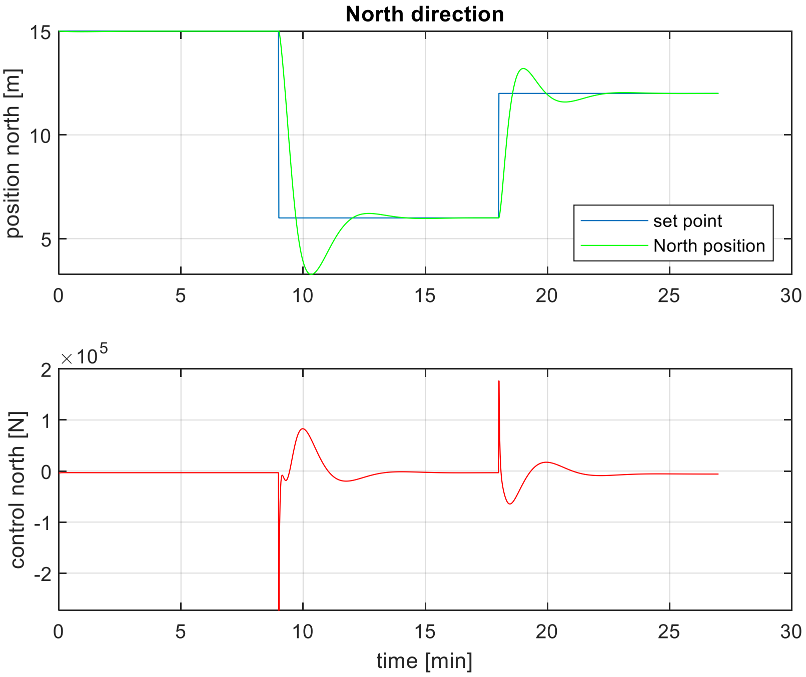


Figure 7 Position and thrust in north axis

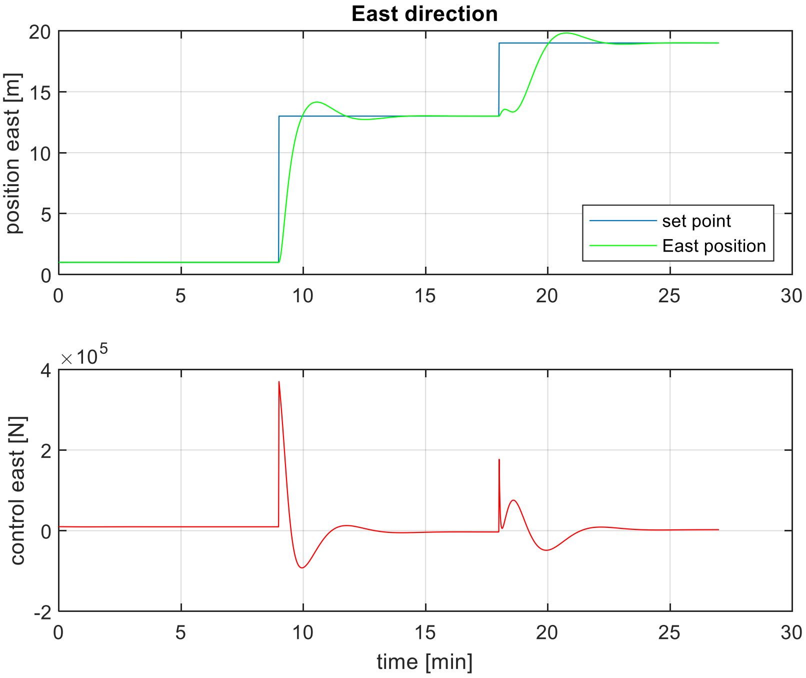


Figure 8 Position and thrust in East axis

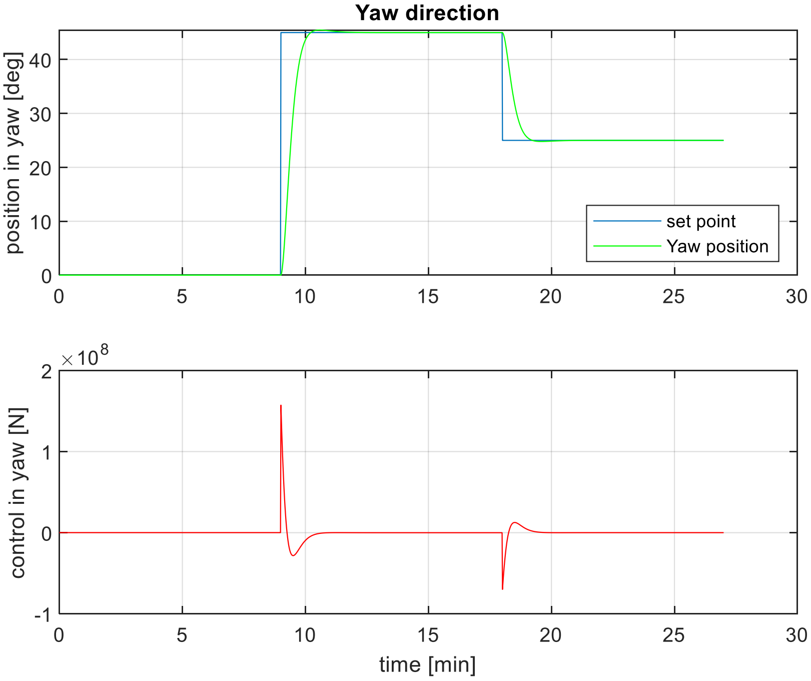


Figure 9 Position and thrust in yaw axis

While these simulations results looks promising, in order to stay faithful to the original work and to get a fair representation of the model behavior the noise must be introduced. Like mentioned in the chapter introduction, σ_1 should be between $0.5\ m$ and $0.7\ m$. That is the only standard deviation mentioned with exact values in the original paper. In order to acquire a rough estimate for the other noise components a comparison is setup with the simulation results given by the original paper. To estimate the low frequency noise (σ_2) the simulations from Balchens paper is recreated, and the results are compared.

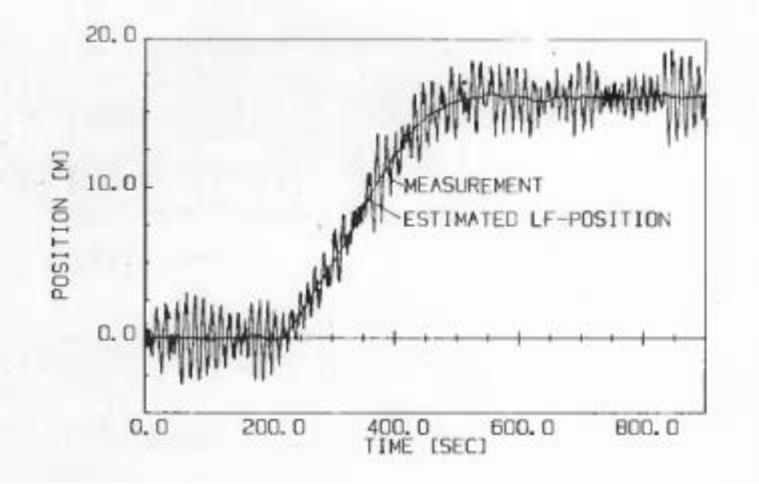


Figure 10 Simulated measurement and LF- estimate of surge

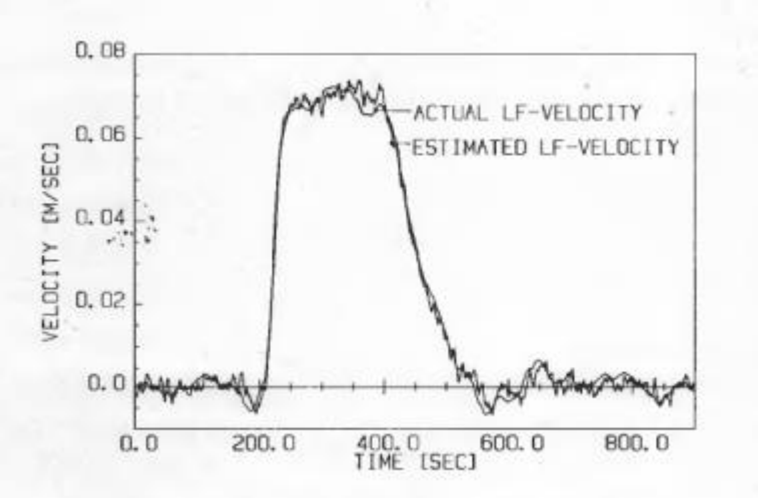


Figure 11 Simulated and estimated LF-velocity in surge

Figure 10 and Figure 11 is connected and show one of the simulation results from the original paper. Figure 10 shows the position, and while a reference value is not shown, it is assumed to show a step change in reference value from 0, to 16 at time 5 minutes. Figure 11 shows the low frequency velocity during the set point change, and Figure 12 shows the same scenario being run by MATLAB with $\sigma_2 = 0.0025$. To arrive at this value was just trial and error until the resulting plots roughly matched in noise frequency and amplitude.

The velocity in the MATLAB simulation spikes to much greater values than in Balchens simulation, and thus the vessel also reaches setpoint faster. The velocity from Balchens simulation showed in Figure 11 seems to reach some limit around $0.07\ m/s$, possible due to strong opposing wind or current. But nether of these factors are explicitly covered for this simulation.

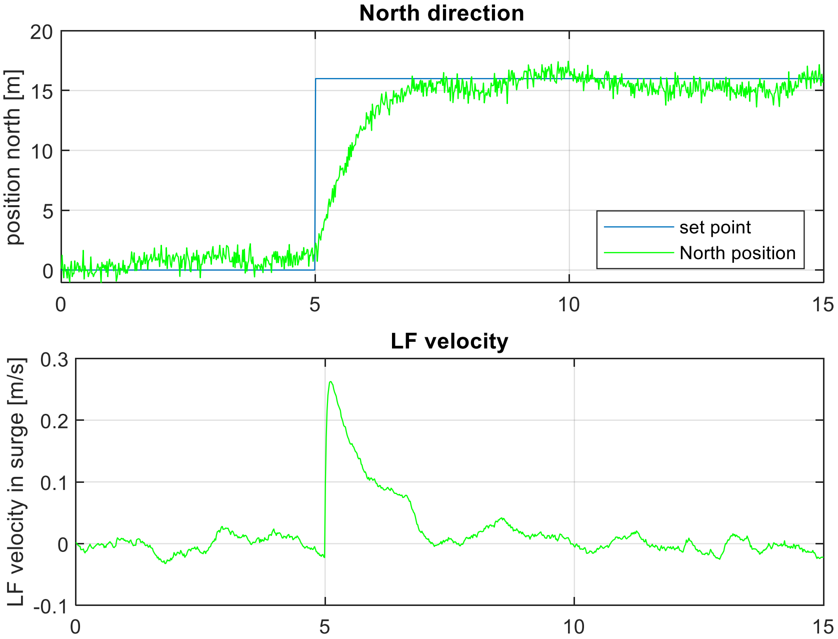


Figure 12 Simulated LF-velocity in surge

When introducing noise to the output model and low frequency model, the simulation results shows that the controller does struggle slightly more. Figure 13, Figure 14 & Figure 15 show the simulation with $\sigma_1 = 0.5\ m$ and $\sigma_2 = 0.0025\ m/s^2$. And this does resemble the expected low frequency noise presented in Balchens paper. The position plotted here is the y state array from the output model.

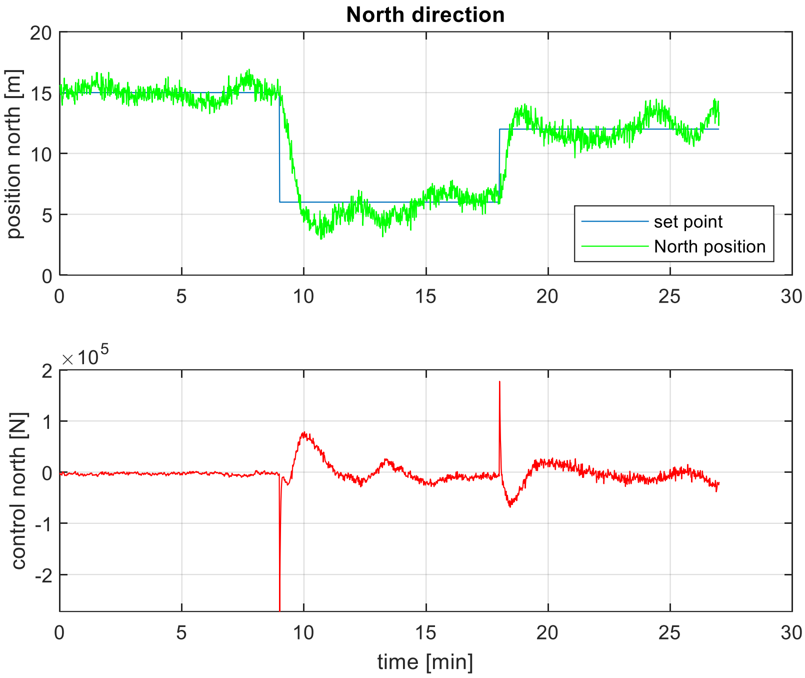


Figure 13 Position and thrust in north axis

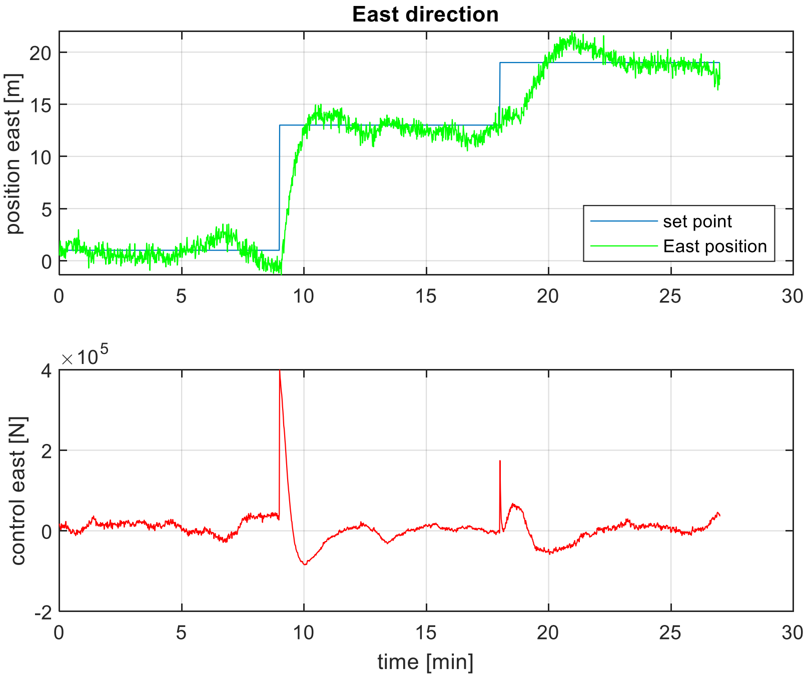


Figure 14 Position and thrust in east axis

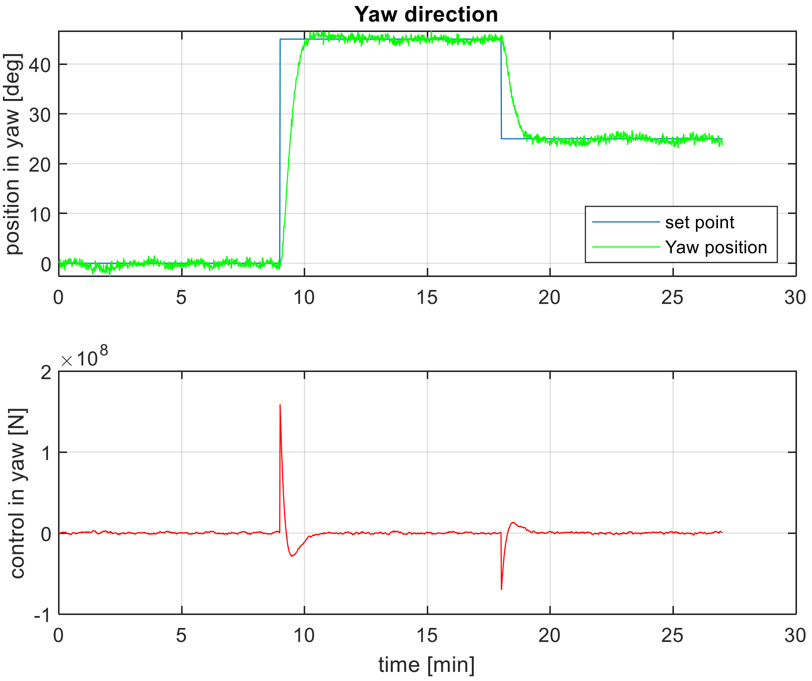


Figure 15 Position and thrust in yaw axis

Balchen does not mention any upper thrust limit for any axis or vessel, but the thrust plots from Balchens paper show thrust peaking at 17 tons force, which would be roughly $1.7 \cdot 10^5\ N$. The MATLAB simulation does not take any thrust limit into consideration but does not produce thrust significantly higher.

To dive deeper into this simulation and see how the observer performs under these noisy conditions, the “actual” simulated position and velocity states is compared to the estimated states in Figure 16 and Figure 17. Unlike the previously plots, these do not show the output model, but rather the internal states from the vessel and controller classes. These state arrays only exist in the program in the vessel coordinate frame, and thus they are not plotted in a NED coordinate frame. For that reason the values in previous output plot does not match the values shown in Figure 17. However, considering that the observer only receives the considerably noisy output signal, it tracks all 6 states rather well.

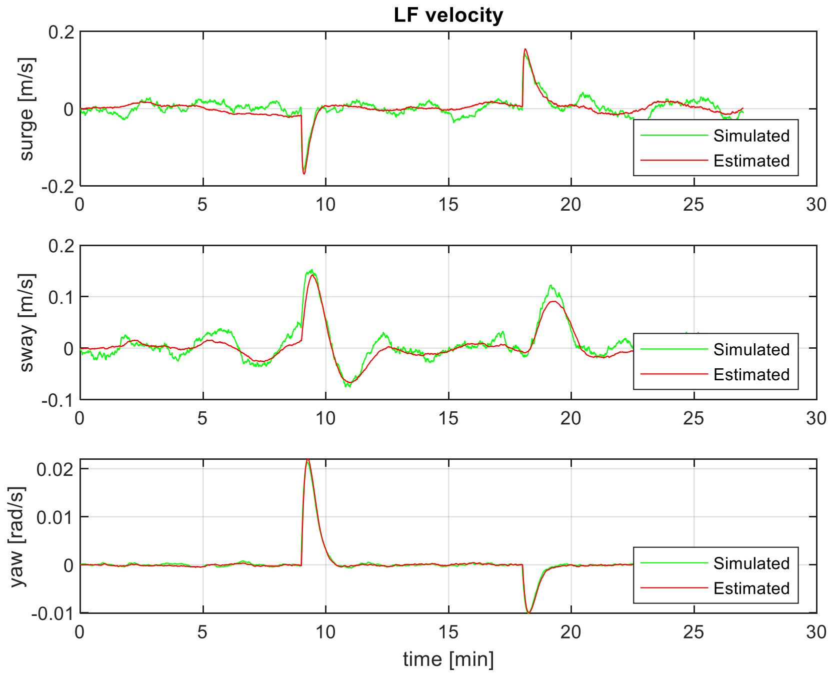


Figure 16 Simulated and estimated position

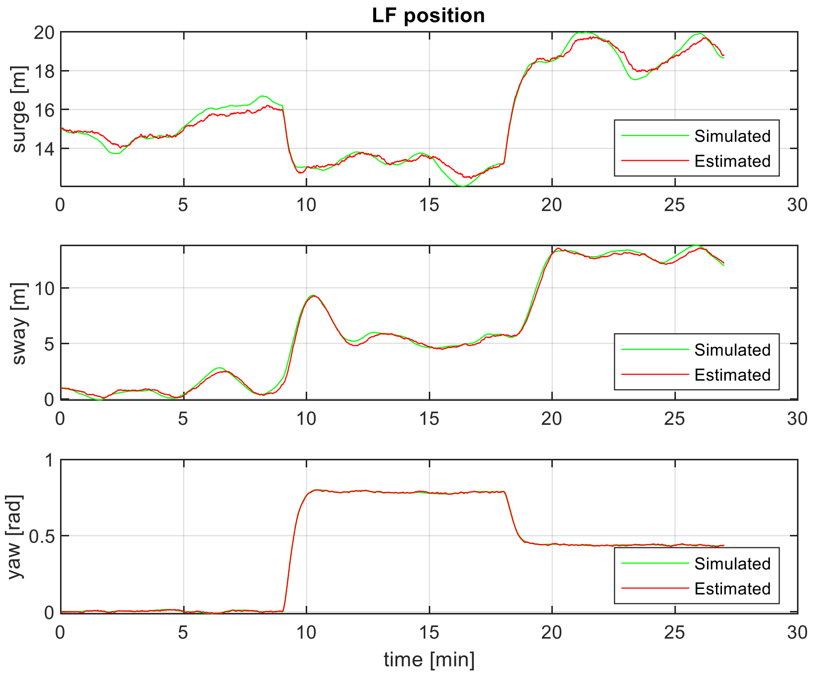


Figure 17 Simulated and estimated velocity

The controller can easily compensate for a larger value of σ_1, the output model noise, but very sensitive to the value of σ_2, the low frequency acceleration noise. Figure 18 shows the simulation results with $\sigma_2 = 0.01$ and while the controller keeps the vessel in the rough area of the set pint, but it fluctuates with $\pm 5m$.

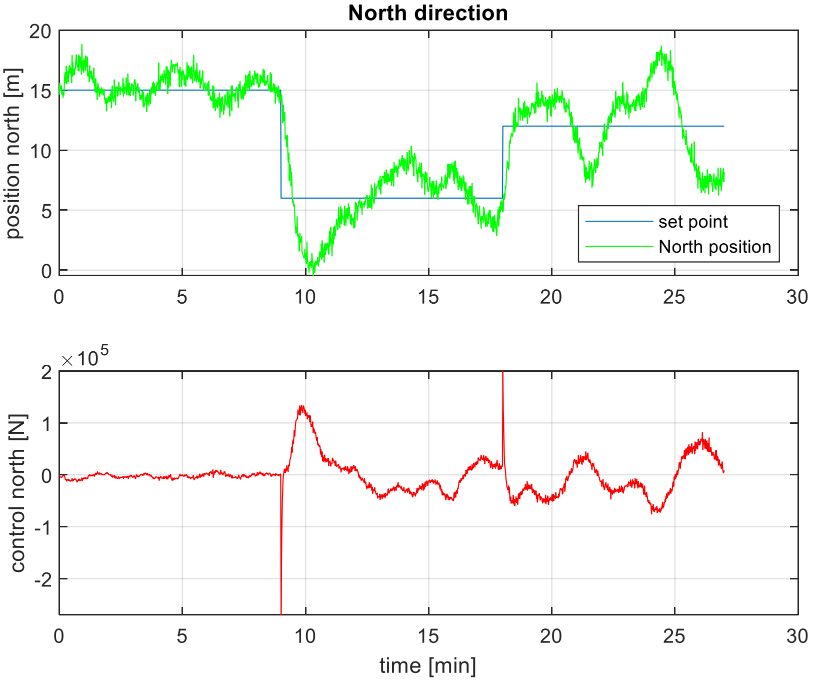


Figure 18 Position and thrust in north axis

5.2 High Frequency model

Because the high frequency model equations are completely independent from the total velocity of the ship and environmental parameters, the results shown in these figures are the high frequency model alone. In order to estimate the correct standard deviation used for omega and the high frequency acceleration equations, a comparison is made against the high frequency plot from Balchens paper shown in Figure 19. This figure shows that the high frequency velocity is expected to sit at roughly 1 cycles pr minute ($1/60\ Hz$), with an amplitude of ~0.4.

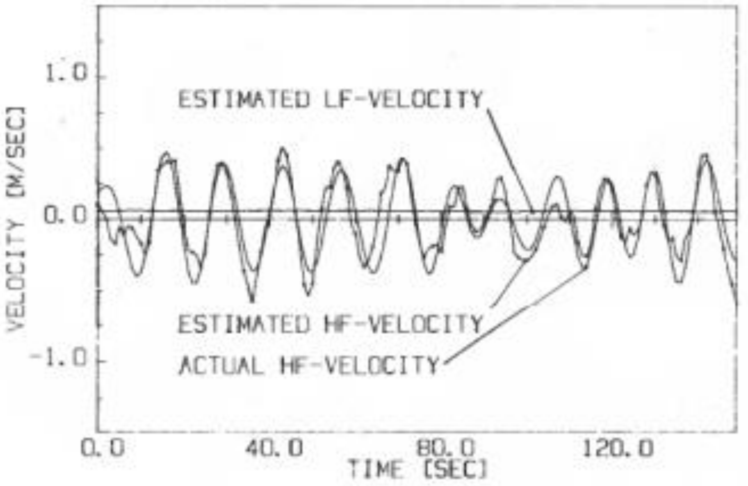


Figure 19 Actual and estimated HF-velocity

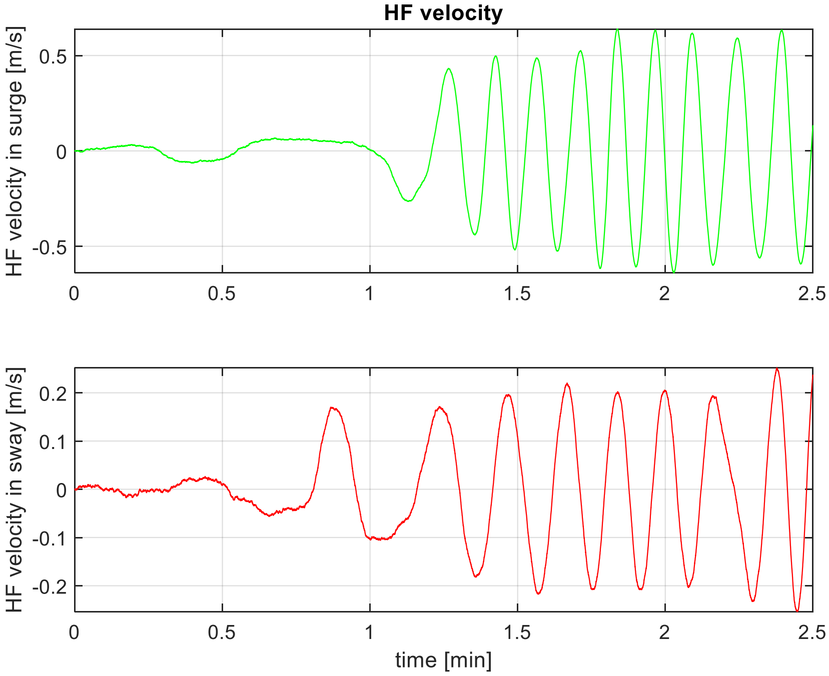


Figure 20 HF-Velocity in surge & sway

Figure 20 shows the simulated HF velocity in both surge and sway under the same timeframe as in Figure 19. These results are achieved with $\sigma_3 = 0.05$ & $\sigma_4 = 0.5$. This simulation was also executed with a timestep of 0.01s, and the reason for this and why it matters will be explored further into the chapter. While the amplitude is in roughly the same range, the frequency here is significantly higher. But because of the random nature of this model, every execution gives a slightly different result. And Figure 21 shows the results of another execution under the exact same parameters as in Figure 20, and the results are noticeably different. Taking into the consideration that an oscillating system of random values will never execute exactly the same, these do look close enough to the target.

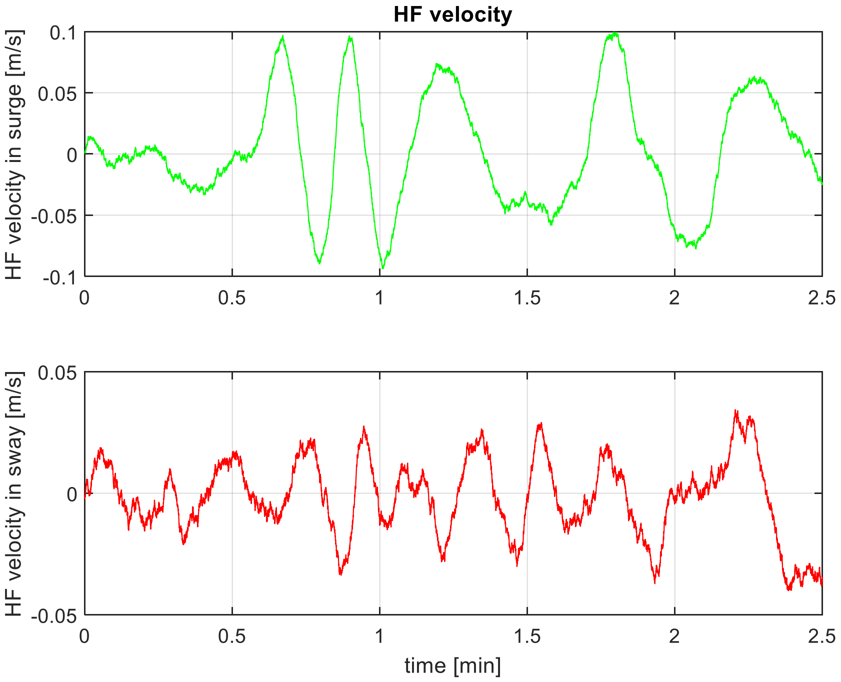


Figure 21 HF-Velocity in surge & sway

But the high frequency model introduces significant problems that becomes apparent when the time is extended. Figure 22 shows the simulation results again with the same parameters, with exception of the time being extended, which illustrates that given enough time the high frequency model oscillates out of control, and appears to be a violently unstable system. Figure 23 shows the values of ω_1 & ω_2 from the same simulation as in Figure 22. These are the frequency used for the high frequency model in surge and sway respectively, and it illustrates the “random walk” nature of these values. It’s not instantly obvious whether or not the values of ω_i is related to the instability of the velocity.

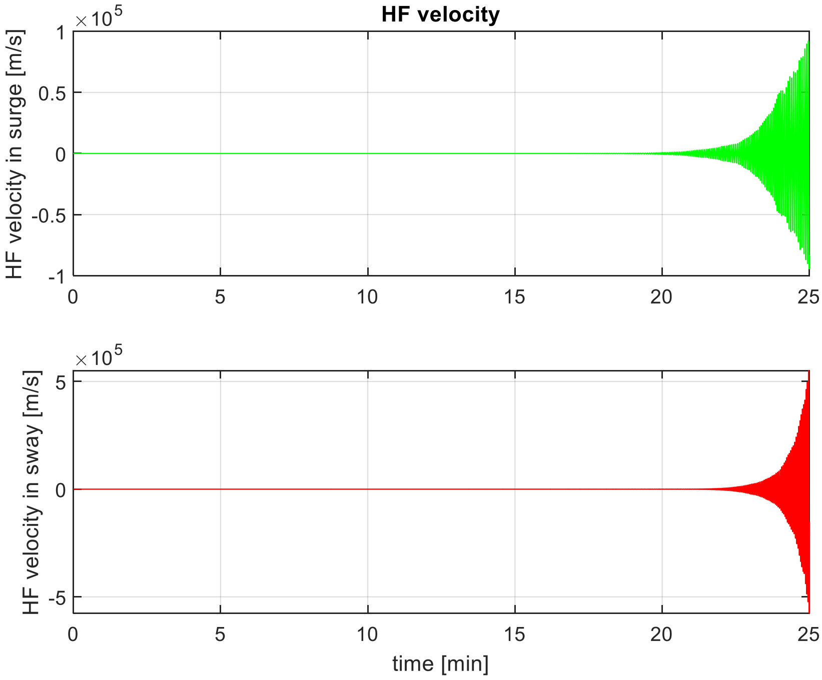


Figure 22 HF-Velocity in surge & sway

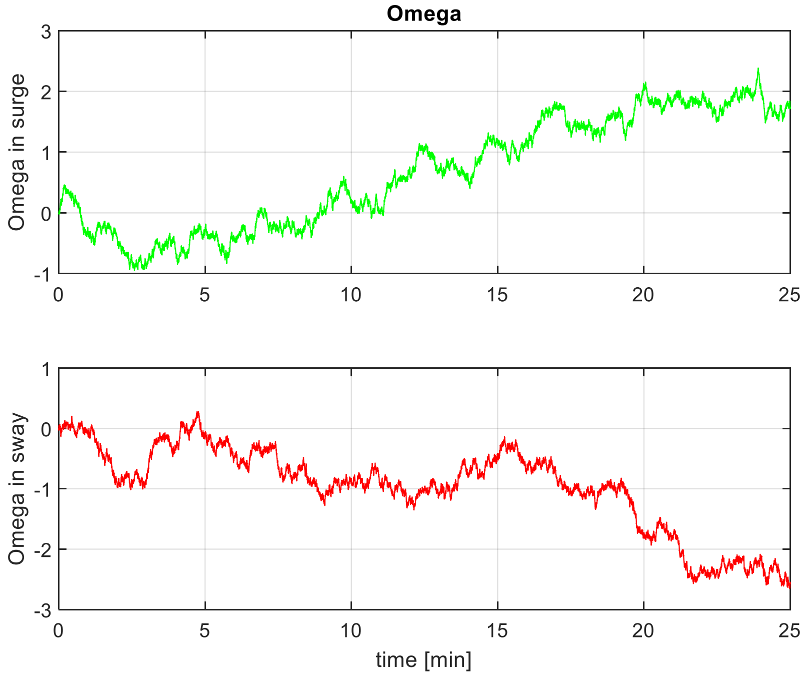


Figure 23 Omega for surge and sway

Rather the choice of time step and method of integrating seems to have a profound effect. Up until this point every differential equation has been integrated with Eulers forward method, and as mentioned all of the high frequency simulations results presented has been simulated with a timestep of 0.01 s. The reason for this deviation from the implied timestep of 1 second is that the results presented in Figure 20 would show the instability even on that small timescale. In an attempt to test if Eulers forward is too simple of an integration method to achieve stability the method was changed to Runge Kuttas 4^{th} order in Figure 24 and Figure 25, and it still appears unstable, but the amplitudes stays within reasonable limits.

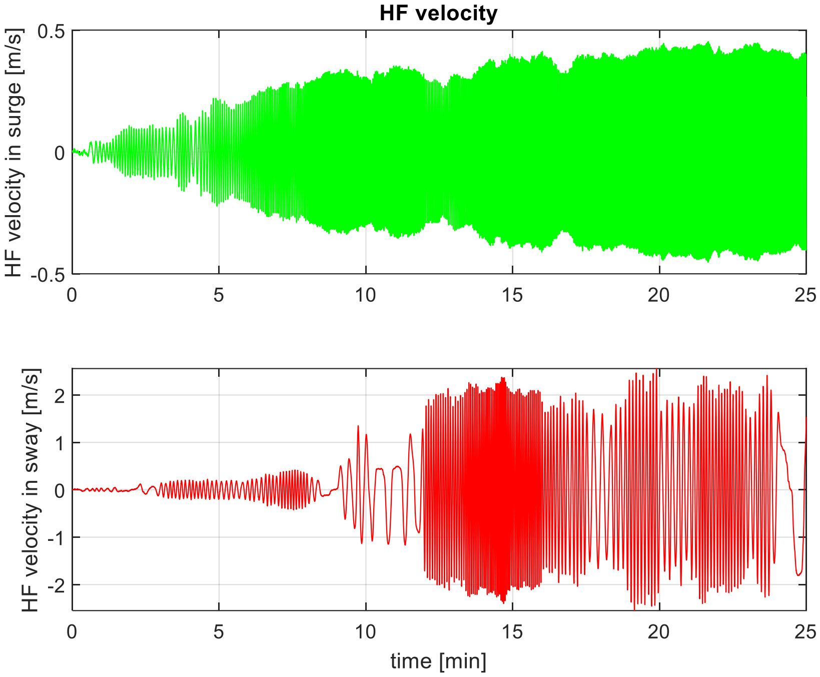


Figure 24 HF-Velocity in surge & sway with runge kutta

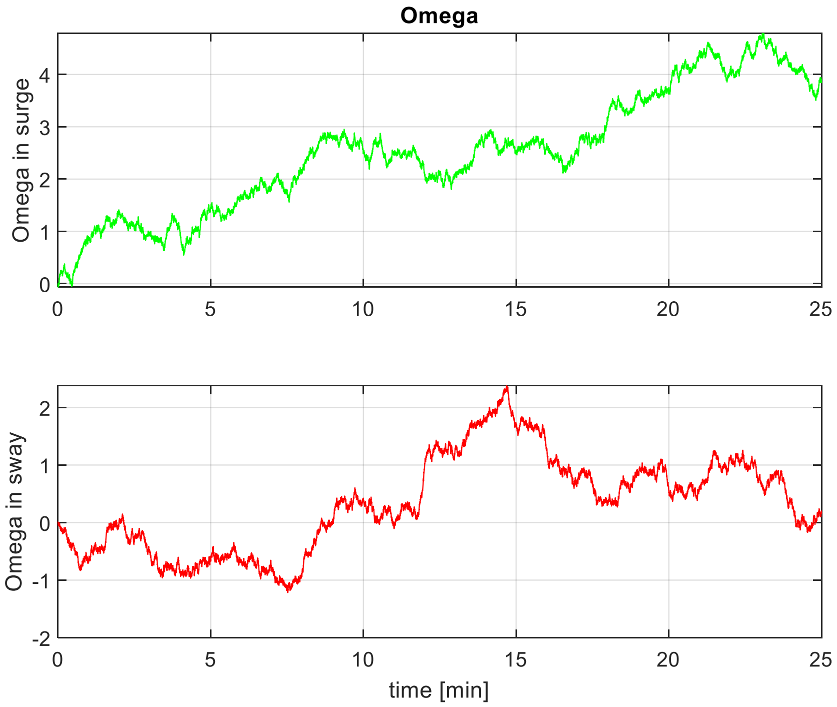


Figure 25 Omega for surge and sway with runge kutta

But when the time scale is again extended as shown in Figure 26 & Figure 27, the violent instability reappears. And just to reiterate, these figures show the simulation of the high frequency model with a timestep of 0.01 s, using runge kutta 4^{th} integration, over a modest timespan of just above 4 hours, and the model still shows to be clearly unstable. Testing also show that applying limits for ω or using a static value does reduce the speed at which the oscillations go out of control, but it's still unstable and given enough time the amplitude explodes. This is also seen in Figure 26 & Figure 27, where omega in surge peeks at 10, and the surge speed becomes unreasonably high at almost 1 km/s, but not exploding. While the sway omega is holding within ± 2 and the sway velocity ends in the order of $2 \cdot 10^{20}\ m/s$.

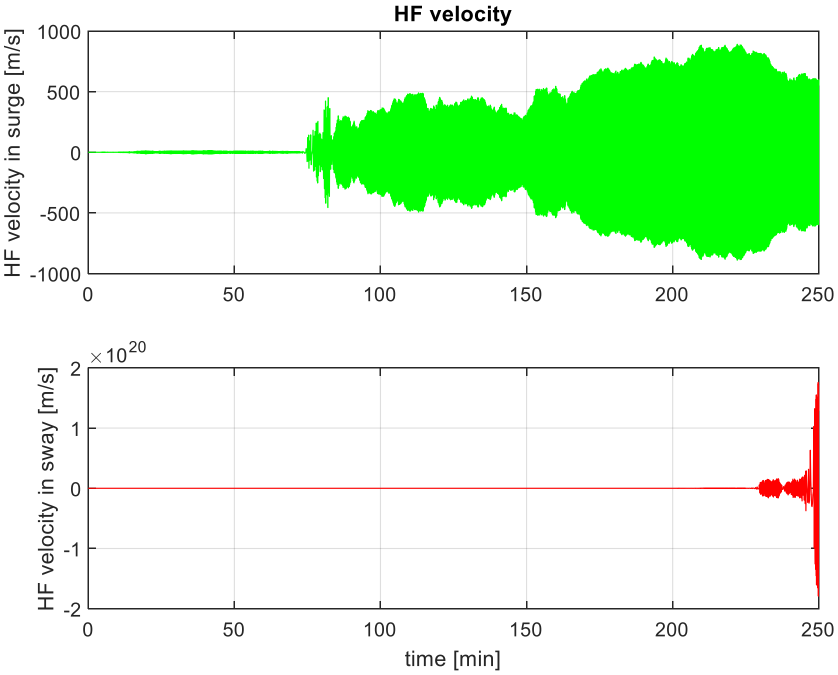


Figure 26 HF-Velocity in surge & sway with runge kutta

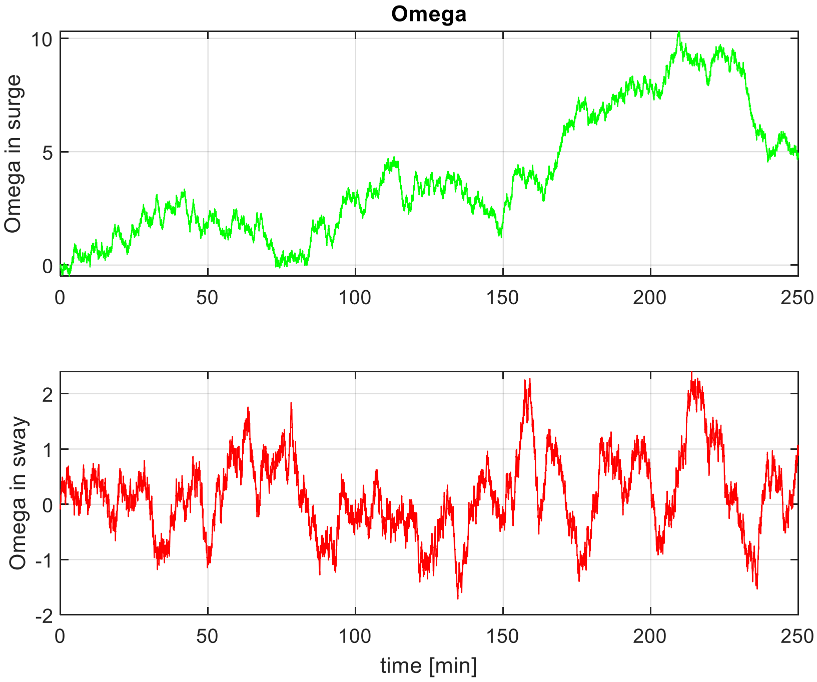


Figure 27 Omega for surge and sway with runge kutta

5.3 Combined model

With the complete system of both low and high frequency models worked together, and Balchens original model simulated in its entirety the problems with the high frequency model become even more apparent. Because of the unstable high frequency model all the plots ends up looking like Figure 28 where one or more values explodes to the point of reaching infinity, causing every formula to start returning either NaN of infinity. This happens even faster here, and it's not directly because the high frequency model produce any "out of bounds" values, but it's enough that the high frequency model produce something significant enough to trough the rest of the feedback loop out, and one problem in one axis perpetuate throughout causing the simulation to halt before it has computed a minute.

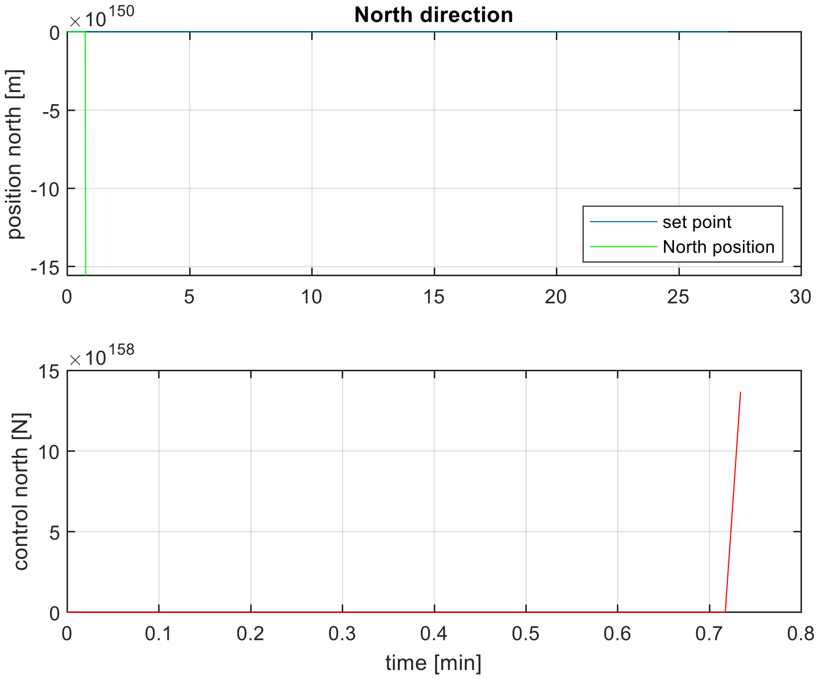


Figure 28 Position and thrust in surge

The only adjustment without altering the original model too much that gives a reasonable output is to set a very low static value for omega. But hampering the high frequency model this much ends up reducing it to such a low contribution it is indistinguishable from the noise already introduced in the low frequency model and output model.

Some attempts where made to rescue the idea of the high frequency model by replacing it with sinus functions, but that caused the low frequency model to start to oscillate, and a change like that to the model would require the observer and possibly the controller to be redesigned to get the wanted control response. Such a large diversion from the original work is out of scope for this paper. Balchens original model can work over a very limited timespan, but because of its unstable behavior it cannot be included, which also takes with it the high frequency observer.

5.4 Simplified Model

The simplified model is the low frequency model but with the drag coefficients set to zero as mentioned in chapter 4. In order to evaluate how representative it is a comparison is setup with both the simplified and non-simplified version. To get a fair comparison the random noise is not included in these simulations. Figure 29 shows the simplified and original model side by side in the same scenario as shown in the start of this chapter. Here the simplified model is the one being controlled, meaning its the outputs of the simplified model that is fed into the controller, while the original model gets the same thrust vector. In this plot it appears that the two models are close to identical, with the two lines seemingly overlapping completely.

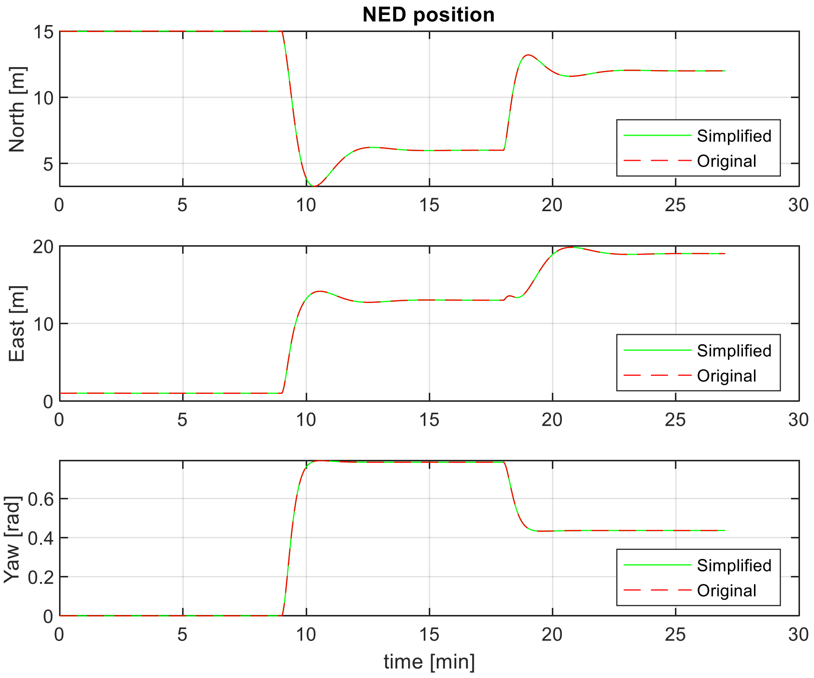


Figure 29 NED position in north, east and yaw directions

To get a better representation of the differences between the simplified and original model the difference between the two are calculated and plotted in Figure 30 for all three axis. Over the 27-minute timespan the offset only accumulates to $4 \cdot 10^{-7}\ m$. The covariance of these offsets has been calculated to $3.2748\ \cdot 10^{-14}$ in north direction, $2.4309\ \cdot 10^{-14}$ in east direction and $9.8027\ \cdot 10^{-23}$ in yaw.

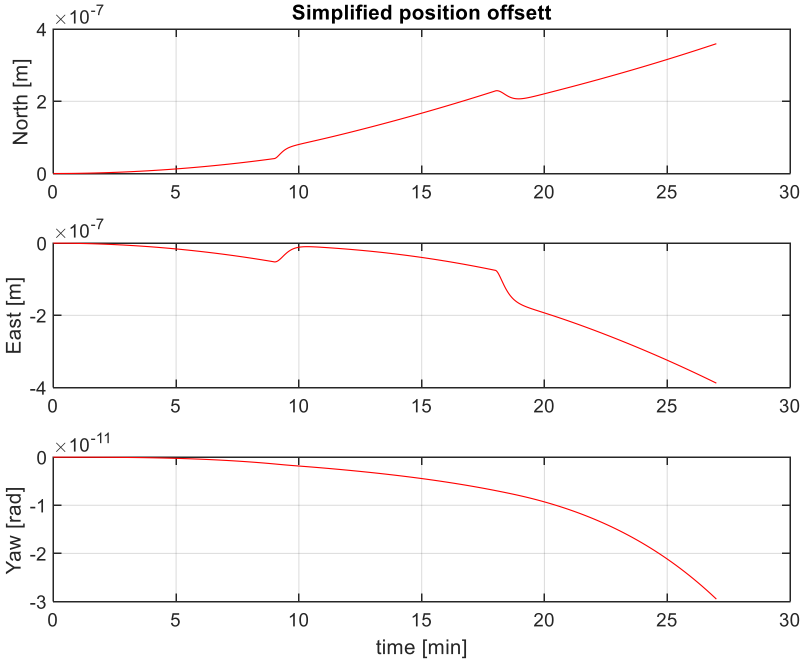


Figure 30 Difference between simplified and original model

These offsets are very small in this 27-minute time window, but they do show a growing trend. Figure 31 shows the same scenario over a much longer timespan, and here the offset grows to a considerable amount. But for the practical purposes of control algorithms these errors would be continuously corrected based on measurements long before the offset reaches any meaningful value.

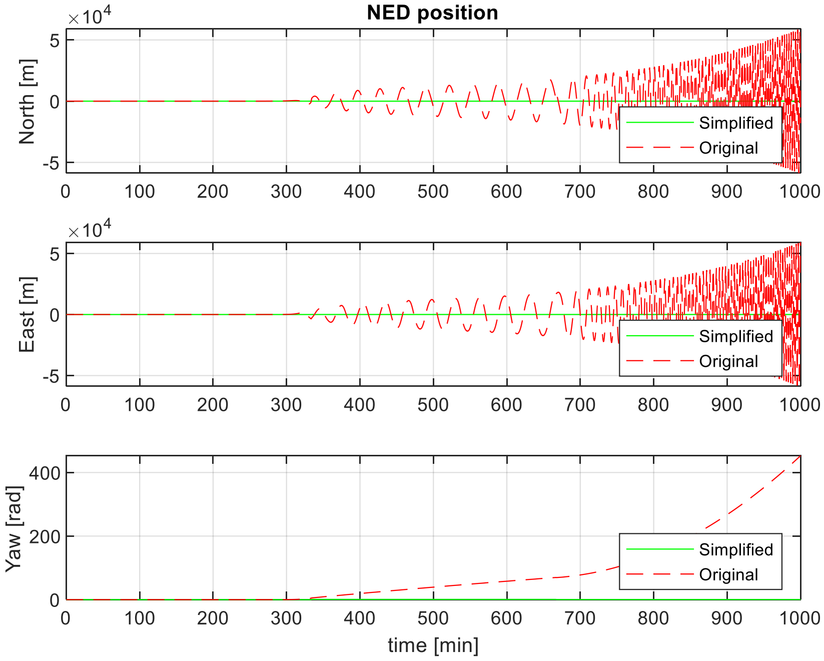


Figure 31 NED position in north, east and yaw directions

6 Conclusion

The dynamic positioning and control theory presented by Balchen in his 1980 paper [1] has successfully been implemented in its entirety within MATLAB. The low frequency model is replicated and behaves like presented in the original paper, but the high frequency model turns out to be very unstable under a long enough timeline. A cause for the unstable behavior has not been found. But both the stability problem and oscillation in the low frequency model was causes for the model changing in Sælids 1983 paper [2].

A possible cause why the high frequency stability problem was not apparent to Balchen in 1980 could have been because computing power was a limiting factor, and he probably did not have the luxury of being able to rerun the simulation several times. Not all executions of the high frequency model led to unstable behavior within the timeframes used, but the vast majority did in at least one of the 3 axis. It's not unthinkable that with limited computing power and very few executions of the simulation, luck would have it that the results looked promising.

On the other hand the simplification of the low frequency model shows to be extremely close to the original. The simplified model is linear and therefor much simpler to analyze and develop control for.

References

[1] Balchen et al 1980
"A dynamic positioning system based on Kalman filtering and optimal control"

[2] Steinar Sælid et al
"Design and Analysis of a Dynamic Positioning System Basen on Kalvamn Filter and Optimal Control"

[3] Nour Mohamad Bargouth
"Dynamic positioning, system identification and control of marine vessels"

[4] David D. Ruscio
"System Theory, StateSpace analysis and Control Theory"

[5] Sjøhistorie "Seaway Swan"

Appendices

Appendix A Master Thesis Task Description

Appendix B MATLAB Code (zip file)

University of South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Dynamic Positioning, system identification and control system of marine vessels

USN supervisor: David Di Ruscio

External partner: zzz

Task background:
One of the first mathematical models used for Dynamic Positioning (DP) of ships was the Balchen et al 1980 model. This model was used for building a DP system based on Kalman filtering and optimal control. It is of interest to reconstruct this DP system and possibly make a modified and simplified model based on three double integrators (putting the drag and momentum coefficients equal zero) in surge, sway and yaw (the three moving directions), and from this make a modified DP system.

Task description:

1. Perform a literature research about DP systems of ships.
2. Implement the modified Balchen et al DP system in MATLAB or similar.
3. Perform simulation experiments.

Student category: IIA students

Is the task suitable for online students (not present at the campus)? Yes

Practical arrangements:
zzz

Supervision:
As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor (date and signature):

Student (write clearly in all capitalized letters): 223787 Kim Hoftvedt

Student (date and signature):